Mapping quantum circuits to shallow-depth measurement patterns based on graph states

IOP Publishing Quantum Sci. Technol. 10 (2025) 015010 https://doi.org/10.1088/2058-9565/ad802b

Quantum Science and Technology

OPEN ACCESS

RECEIVED
21 December 2023

REVISED
2 September 2024

ACCEPTED FOR PUBLICATION
26 September 2024

PUBLISHED
15 October 2024

PAPER

Mapping quantum circuits to shallow-depth measurement patterns based on graph states

Thierry N Kaldenbach[1,3,4,*] and Matthias Heller[1,2,4]

[1] Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany
[2] Technical University of Darmstadt, Interactive Graphics Systems Group, Darmstadt, Germany
[3] German Aerospace Center (DLR), Institute of Materials Research, Cologne, Germany
[4] Both authors contributed equally to this work.
[*] Author to whom any correspondence should be addressed.

E-mail: thierry.kaldenbach@dlr.de and matthias.heller@igd.fraunhofer.de

Keywords: quantum computing, quantum simulation, measurement-based quantum computing, quantum algorithms, dynamic circuits, graph states, stabilizer formalism

Abstract

The paradigm of measurement-based quantum computing (MBQC) starts from a highly entangled resource state on which unitary operations are executed through adaptive measurements and corrections ensuring determinism. This is set in contrast to the more common quantum circuit model, in which unitary operations are directly implemented through quantum gates prior to final measurements. In this work, we incorporate concepts from MBQC into the circuit model to create a hybrid simulation technique, permitting us to split any quantum circuit into a classically efficiently simulatable Clifford-part and a second part consisting of a stabilizer state and local (adaptive) measurement instructions—a so-called standard form—which is executed on a quantum computer. We further process the stabilizer state with the graph state formalism, thus, enabling a significant decrease in circuit depth for certain applications. We show that groups of mutually-commuting operators can be implemented using fully-parallel, i.e. non-adaptive, measurements within our protocol. In addition, we discuss how groups of mutually commuting observables can be simulatenously measured by adjusting the resource state, rather than performing a costly basis transformation prior to the measurement as it is done in the circuit model. Finally, we demonstrate the utility of our technique on two examples of high practical relevance—the Quantum Approximate Optimization Algorithm and the Variational Quantum Eigensolver (VQE) for the ground-state energy estimation of the water molecule. For the VQE, we find a reduction of the depth by a factor of 4 to 5 using measurement patterns vs. the standard circuit model. At the same time, since we incorporate the simultaneous measurements, our patterns allow us to save shots by a factor of at least 3.5 compared to measuring Pauli strings individually in the circuit model.

1. Introduction

Measurement-based quantum computing (MBQC) offers an interesting alternative to the standard circuit model. While unitary operations in the circuit model are realized by sequential application of quantum gates, MBQC operates on a highly entangled state, called resource state, on which unitary operations can be implemented via adaptive measurements [1–3]. Some of these measurements can be performed in parallel, which leads to a compelling feature of MBQC: the parallel application of unitaries, which in the gate model would be applied sequentially.

Due to its universality, it is possible to map any quantum circuit to the MBQC model and vice versa. Forward and backward translation between the circuit model and MBQC can lead to beneficial tradeoffs in terms of depth and space complexity [4, 5]. Different techniques for translation between these models [6, 7] and optimization of adaptive measurement patterns [8] have been studied based on graph-theoretical tools

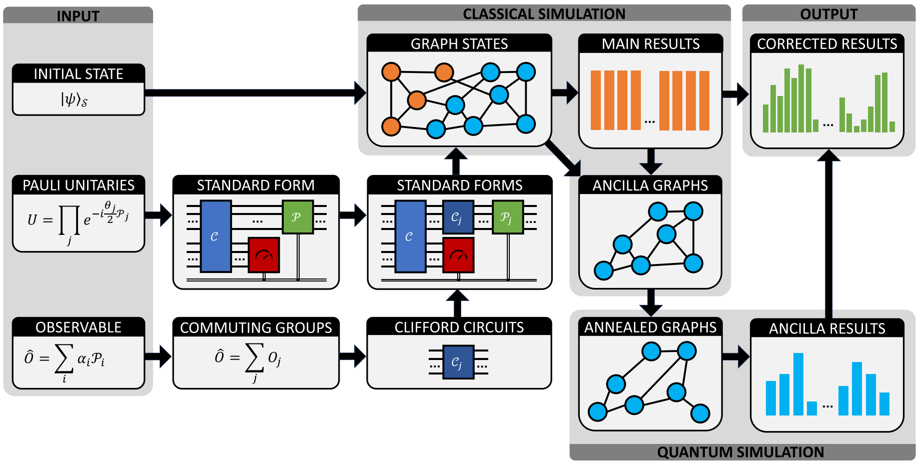


Figure 1. Overview of our compilation pipeline to map a given unitay U composed of several Pauli unitaries into a shallow-depth measurement-pattern based on optimized graph states.

such as causal flow [9], generalized flow (gflow) [7, 10, 11] and Pauli flow [12]. Recently, approaches employing ZX-calculus [13] have also attracted interested in the context of circuit optimization [14] and conversion [7, 12, 15].

In this paper, we introduce a straight-forward algorithm that allows the mapping of a given quantum circuit to a graph state, which, together with local Clifford operations and measurement instructions, allows to perform quantum computations within the MBQC model.

The starting point of our circuit-to-graph conversion is a list of n Pauli unitaries—operators of the form $e^{-i\theta\mathcal{P}}$, where $\mathcal{P}$ is a Pauli string—that act on some set of qubits, which we call main qubits in the following. We first derive measurement patterns for each Pauli unitary individually—so called Pauli gadgets [16, 17]—by introducing one additional ancillary qubit for each unitary. We then concatenate all n patterns to a single one and reduce its depth by bringing it into standard form by shifting all corrections to the end of the computation [4, 18]. We show that using this standard form, it is possible to apply commuting Pauli unitaries in parallel. In its standard form, the pattern consists of three parts: a Clifford layer, a measurement layer and a final Pauli correction acting on the main qubits.

Given an initial (stabilizer) state of the main qubits together with some Clifford circuit that can be attached to the end of the pattern to perform simultaneous measurements of a set of commuting observables [19, 20], we then use a graph-state simulator [21] to map the unitary to a graph state. Next, we classically simulate the measurement of the main qubits, such that we are left with a measurement distribution of these and a graph state acting on the ancilla qubits only. Before we finally run the circuits corresponding to the patterns on these ancilla graphs on a quantum computer, we reduce their preparation depth using simulated annealing [22, 23]. The full compilation pipeline of our algorithm is depicted in figure 1.

We use our algorithm in two scenarios, which are often discussed as near-term applications for noisy intermediate-scale quantum (NISQ) devices [24]: the Variational Quantum Eigensolver (VQE) [25, 26] in the context of molecular simulations and the Quantum Approximate Optimization Algorithm (QAOA) [27, 28] in the context of binary optimization problems. In particular, for the ground-state energy calculation of H_2O, we demonstrate that our algorithm achieves an efficient mapping of the qubit-ADAPT-VQE [29] to highly shallow circuits. Using this example, we showcase another important property of our mapping technique, namely, that groups of mutually-commuting observables can be measured simultaneously [20] without significantly increasing the circuit depth. This is due to fact that we can absorb the appropriate basis transformation into the underlying graph state, unlike in the quantum circuit model. The reduction in circuit depth advances the practical utility of quantum computation in the NISQ-era where circuit depth is strictly limited by coherence time. Meanwhile, the importance of our results also holds for fault-tolerant quantum computation [30], where the speedup due to the parallel application of quantum operations is still desirable.

The remainder of this paper is structured as follows: section 2 gives an overview over related work and highlights our new contributions compared to these. Section 3 provides an overview on the graph state formalism (section 3.1) and MBQC (section 3.2). Based on these fundamentals, we proceed to introduce our algorithm step by step in section 4. We start by reviewing common circuit structures for Pauli exponentials

[31, 32] and combine them with the One-Way Quantum Computer (1WQC) protocol [1, 33–35] in section 4.1, revealing a general circuit structure consisting of Clifford, measurement and Pauli parts. In section 4.2, we elaborate how commuting Pauli strings can be implemented through parallel measurements. Section 4.3 details how the Clifford structure may be implemented with constant circuit depth. Next, in section 4.4 we propose a hybrid simulation scheme, entailing a classical simulation of the main register and a quantum simulation of the ancilla register. For practical feasibility, in section 4.5 we also provide an optimization routine for the underlying graph states to minimize the number of required entangling gates. Finally, we show several applications demonstrating the utility of our algorithm in section 5, more specifically the QAOA in section 5.1 and the VQE in section 5.2. Furthermore, we give an outlook in section 6 and show some technical details in appendices A–G.

2. Related work and our contributions

To the best of our knowledge, the first time when MBQC has been used to reduce the depth of quantum circuits by translating the circuit into a pattern and employing parallelization was done in the work by Broadbent and Kashefi [4]. Their translation scheme works by transpiling the circuit to the set of CZ and $HR_z(\theta)$ gates first. For the case of Pauli unitaries (i.e. unitaries corresponding to exponentials of Pauli strings, cf equation (5)) which we consider in this work, this scheme is not optimal in the sense that it introduces more ancillary qubits than necessary (which of course can be removed again by classical simulation).

Chan *et al* [17] consider so-called Pauli gadgets, which implement unitaries of the form as in equation (5) using one ancillary qubit. The same pattern has also been discussed by Cowtan *et al* [16] in the context of the *ZX*-calculus [7, 36], where it has been used to optimize quantum circuits by reducing the number of required two-qubit gates. While Chan *et al* use Pauli gadgets in a hybrid scheme to develop circuits with a reduced number of two-qubit gates, Cowtan *et al* use them only in intermediate stages of a transpilation rule to simplify a given circuit. However, both these references do not discuss the possibility to construct measurement patterns, which implement several commuting Pauli unitaries in parallel with only one round of measurements. Essentially, due to the commutativity of a given set of (commuting) Pauli unitaries, one can find a standardized form of the combined patterns, which has minimal causal depth, by maximally delaying the corrections in the pattern, see Mhalla and Perdrix [18] and Broadbent and Kashefi [4]. For the case of Pauli gadgets, we discuss this in detail in sections 4.2 and 4.3.

In section 4 we design measurement patterns by transforming Pauli unitaries starting from circuits in the *star* or *star+ancilla* layout (cf figure 3) using ideas from teleportation-based quantum computing [37]. Furthermore, we show that these patterns are equivalent to the Pauli gadgets considered in [16, 17]. However, our construction based on the *star*-layout can lead to patterns with less two-qubit gates compared to the standard gadgets, as we will show in section 5.2.1 for the example of an excitation operator from unitary-coupled-cluster theory.

As described in the Introduction, we use the patterns to map circuits to graph states by introducing one ancillary qubit for each Pauli unitary. A closely related circuit transpilation approach has recently been explored by Vijayan *et al* [38], where circuits are first transformed into an inverse Initialization-CNOT-Measurement form [39–41] via T-state injection [31], followed by gate teleportation of T gates to separate the Clifford structure, which is then mapped to a graph state. In a subsequent work [42], this approach was used to estimate the fault-tolerant resources required for several algorithms. A similar hybrid approach for Clifford+T circuits using Pauli-based computation [43] rather than graph states is presented in [44].

A novelty of our approach is the incorporation of Clifford circuits that allow the parallel measurement of commuting observable into our pattern by absorbing them into the Clifford structure (or the graph state) of the pattern. In the circuit model, these Clifford circuits lead to a significant increase in circuit depth. For the simulation of H_2O using qubit-ADAPT we found that this procedure did not lead to more complicated graph states when optimizing them using simulated annealing to reduce the number of edges.

3. Preliminaries

To set the stage for our circuit-to-graph-state conversion algorithm, we first review some basic concepts needed to understand the idea. We start by reviewing the definition of graph states and their connection to stabilizer states and then give a brief introduction to MBQC.

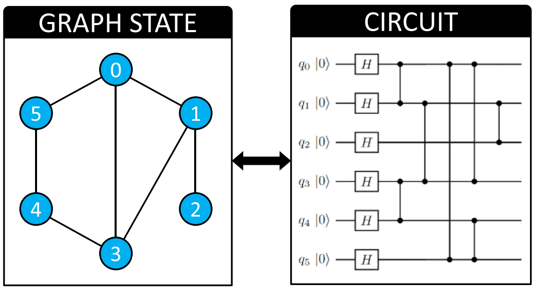


Figure 2. A graph state (left) and the corresponding quantum circuit (right). Some of the entangling gates can be applied in parallel, hence the total depth of the circuit in terms of entangling layers is only three.

3.1. Graph- and stabilizer states

The measurement patterns obtained through our protocol are based on a computational resource state called graph state [21, 45, 46]. An N-qubit graph state $|G\rangle$ is associated to an undirected graph $G = (V, E)$, whose $|V| = N$ vertices correspond to N qubits prepared in the $|+\rangle \equiv 1/\sqrt{2}(|0\rangle + |1\rangle)$ state, while the set of edges E describes the action of controlled-Z (CZ) operations among them. It can therefore be constructed as

$$|G\rangle = \left(\prod_{a,b\in E} CZ_{ab}\right)\left(\prod_{a\in V} H_a\right)|0\rangle^{\otimes N}. \tag{1}$$

The action of all CZ_{ab} gates commute with each other and can, in principle, be applied in parallel. In order to prepare a graph state on a quantum computer (using the circuit model), we thus need a maximum depth of

$$d = \max_{\alpha\in V} |N_G(\alpha)| + 1, \tag{2}$$

where $N_G(\alpha)$ denotes the set of vertices connected to the vertex α, i.e. $|N_G(\alpha)|$ is the degree of α [4, 47]. An example of a graph state together with its corresponding quantum circuit is shown in figure 2.

Equation (1) shows that any graph state on N qubits can be generated through a sequence of Hadamard and CZ gates. Both belong to the Clifford group $\mathcal{C}_N$, which is defined as

$$\mathcal{C}_N = \left\{U \in \mathrm{SU}\left(2^N\right) \mid UPU^\dagger \in \mathcal{P}_N \,\forall P \in \mathcal{P}_N\right\}, \tag{3}$$

where $\mathcal{P}_N$ denotes the N-qubit Pauli group. Any Clifford operator can be generated by three elementary gates (see, e.g. [31]): the Hadamard gate (H), the phase gate (S) and the two-qubit CZ or CNOT gate.

An N-qubit stabilizer state is a quantum state, that can be prepared by a sequence of Clifford gates acting on the $|0\rangle^{\otimes N}$ state. Thus, by definition, every graph state is a stabilizer state—the reverse is not true. However, it can be shown that every stabilizer state is local Clifford equivalent to a graph state (LC-equivalence), i.e. for every stabilizer state a graph state can be found, that can be transformed to the stabilizer state using one-qubit Clifford gates only [48, 49].

Quantum circuits consisting only of Clifford gates can be simulated efficiently on a classical computer according to the Gottesman–Knill theorem [50]. Using the LC-equivalence of stabilizer states with graph states, we can simulate an N-qubit Clifford circuit using $\mathcal{O}(N \ln N)$ space in computer memory. The core idea is to store the graph together with the local Clifford operations (also called vertex operators or VOPs) for each qubit [21].

3.2. MBQC

In the model of MBQC, quantum computations start from a highly entangled many-qubit state, called resource state, which is modified by applying a sequence of adaptive measurements onto a subset of qubits. At first sight, it might seem counter-intuitive that universal quantum computation can be performed using irreversible, destructive measurements. While MBQC involves a loss of information concerning the entire resource state, it still performs unitary transformations on the subset of qubits that are not measured during the computation. On these unmeasured qubits any unitary operation can be implemented, provided the resource state is sufficiently complex.

Although different MBQC schemes exist, here we focus on the so-called cluster model or 1WQC of Raussendorf and Briegel [1, 33–35]. A review of 1WQC and other measurement-based schemes can be found in [2].

In the 1WQC model, all quantum gates are implemented as a sequence of single-qubit measurements on a suitable large cluster state [3]. The measurement basis needed for universal quantum computing is given by

$$M(\theta) = \left\{|0\rangle \pm e^{i\theta}|1\rangle\right\}. \tag{4}$$

The measurement in the $M(\theta)$-basis is achieved by applying the unitary $HR_z(\theta)$ to the computational basis and performing the usual Z measurement.

4. Mapping circuits to graph states

In this section, we introduce an algorithm, that allows the mapping of a quantum circuit to a graph state, that can then be used within the MBQC protocol. The core idea is to map unitaries of the form

$$U = e^{-\frac{i}{2}\theta\mathcal{P}}, \tag{5}$$

where $\mathcal{P}$ denotes an N-qubit Pauli string $\mathcal{P} \in \{I, X, Y, Z\}^{\otimes N}$, to an ancilla qubit in the circuit. To implement such operators in the circuit model, the Pauli string is diagonalized by applying local Clifford operators using the identities $X = HZH$ and $Y = SHZHS^\dagger$. This effectively reduces the operator pool to $\mathcal{P}' \in \{I, Z\}^{\otimes N}$.

We review two common circuit structures representing $\exp(-\mathrm{i}/2\theta\mathcal{P}')$. In the *star+ancilla* layout (figure 3(a)), all non-identity qubits of the string are directly entangled with an ancilla qubit (star-like structure), which is initialized in state $|0\rangle$ and on which the R_z-rotation is carried out. By repeating the same entanglement structure, the entanglement with the ancilla qubit is undone. The star-like entanglement with the ancilla can be interpreted as a computation of the parity of the N qubits in a classical manner. For an even parity, a phase shift of $\exp(-\mathrm{i}\theta/2)$ is applied, otherwise it is $\exp(\mathrm{i}\theta/2)$. Finally, the parity is uncomputed, erasing the ancilla and leaving it in the $|0\rangle$ state [31].

The second structure, which we refer to as the *star* layout (figure 3(b)), works similarly, with the key difference that the entanglement and R_z gate is performed with respect to one of the non-identity qubits instead of an ancilla. Both circuits are equivalent and have their own benefits depending on the problem at hand. Further, it should be mentioned that both circuits can be equivalently realized using a ladder-like entanglement structure, though this approach is not further discussed within our work due to less convenient gate cancellation properties [32].

4.1. Measurement patterns for Pauli unitaries

The first ingredient in converting a given quantum circuit into a graph state is to replace all single-qubit rotations $R_z(\theta)$ by a measurement pattern by introducing one ancilla qubit. Consider the example of a single qubit in an arbitrary state $|\psi\rangle$ entangled with a second qubit in the $|+\rangle$ state via a CZ gate. After measuring the first qubit in the $M(\theta)$ basis, the second qubit is left in the state

$$|\psi\rangle' = X^s HR_z(\theta)|\psi\rangle, \tag{6}$$

where $s \in \{0, 1\}$ is the measurement outcome on the first qubit. Acting with X^s and then with H on the second qubit yields the deterministic final state $R_z(\theta)|\psi\rangle$, which is the desired $R_z(\theta)$ gate. The quantum circuit performing this operation is shown in figure 4 and can be used as a pattern to replace any R_z gate in a given circuit.

Next, we use this pattern to rewrite unitaries defined by equation (5) in the MBQC protocol. As an example, let us consider the unitary $\exp(-\mathrm{i}/2\theta Z_0 Z_1)$. We use the *star+ancilla* layout as starting point to exemplify some aspects of our algorithm. It is straightforward to derive the pattern for the same example in the *star* layout. By replacing the R_z gate on the ancilla qubit with the pattern, we derive the circuit shown in the right panel of figure 4.

So far, it appears that there is no benefit from replacing the R_z gate with the MB-protocol. It is rather the opposite: one more CZ gate and an additional measurement are required to realize the same operation. However, by shifting the classically-controlled Pauli corrections across the Clifford gates to the end of the circuit, the quantum circuit can be separated into three components: a pure Clifford layer, a measurement layer and a correction layer, as shown in figure 5. In accordance with Broadbent and Kashefi [4], we call this the standard form of a pattern.

The standard form of a pattern can be easily achieved by employing the following identities for single-qubit

$$\begin{aligned} HX &= ZH, & HZ &= XH, \\ S^{(\dagger)}X &= YS^{(\dagger)}, & S^{(\dagger)}Y &= XS^{(\dagger)}, \end{aligned} \tag{7}$$

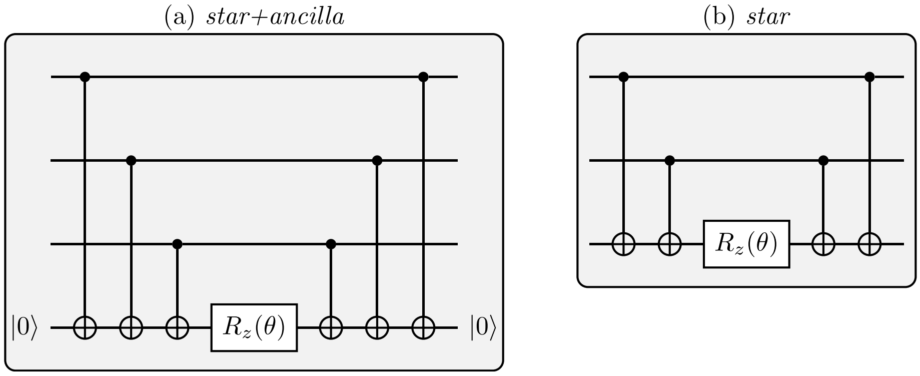


Figure 3. Implementations of the operator $\exp\left(-\mathrm{i}/2\theta Z_0 Z_1 Z_2\right)$ using (a) the *star+ancilla* layout, and (b) the pure *star* layout. The term *star* refers to the star-shaped structure of the entangling gates.

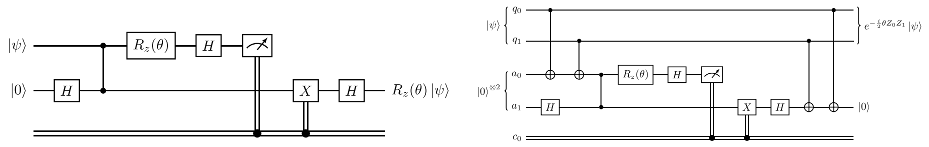


Figure 4. Left panel: Measurement pattern to implement the R_z gate. After measurement of the first qubit, the second qubit is in state $X^s H R_z(\theta)|\psi\rangle$. Right panel: Circuit implementation of the unitary $\exp\left(-\mathrm{i}/2\theta Z_0 Z_1\right)$ using the *star+ancilla* layout, after insertion of the pattern. The ancilla qubit a_1 is in the $|0\rangle$-state at the beginning and at the end.

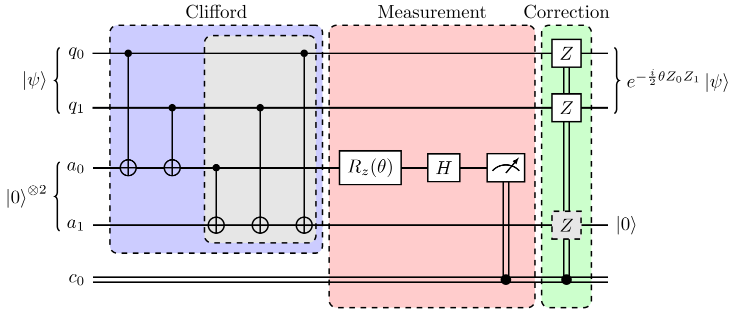


Figure 5. Pattern to implement $\exp\left(-\mathrm{i}/2\theta Z_0 Z_1\right)$ acting on qubits q_0 and q_1, after shifting the Pauli corrections to the end. The gates highlighted in gray arise straightforwardly from modifying the circuit in figure 4 but have no effect on the final state and can thus be left out.

and two-qubit Clifford gates

$$\begin{aligned}
CX_{12}\left(I_1 \otimes Z_2\right) &= \left(Z_1 \otimes Z_2\right) CX_{12} \\
CX_{12}\left(X_1 \otimes I_2\right) &= \left(X_1 \otimes X_2\right) CX_{12} \\
CZ_{12}\left(I_1 \otimes X_2\right) &= \left(Z_1 \otimes X_2\right) CZ_{12}.
\end{aligned} \tag{8}$$

All remaining identities can be directly obtained from $Y \propto Z \cdot X$ and $CZ_{12} = CZ_{21}$.

Interestingly, the ancilla qubit a_1 introduced to perform the gate teleportation on the ancilla qubits a_0 has no effect on the fully modified pattern shown in figure 5. This is a general feature for a pattern in the *star+ancilla* layout. The gray-highlighted right-hand side of the CNOT structure does nothing but compute and uncompute the parity of the main qubits (this is because a_0 already stores the parity), thus leaving a_1 in state $|0\rangle$. Note, that after removing the three gray-highlighted CNOT gates in figure 5, we end up with a pattern that is equivalent to the Pauli gadgets considered in [16, 17].

The last classically-controlled Z gate on the last ancilla can thus always be neglected, since $Z|0\rangle = |0\rangle$. Note that the removal of the right-hand side CNOT structure is not possible if one uses the pure *star* layout instead.

Based on this example, we now introduce measurement patterns for the application of an arbitrary Pauli exponential. We already know that Pauli corrections are applied to all qubits corresponding to non-identity

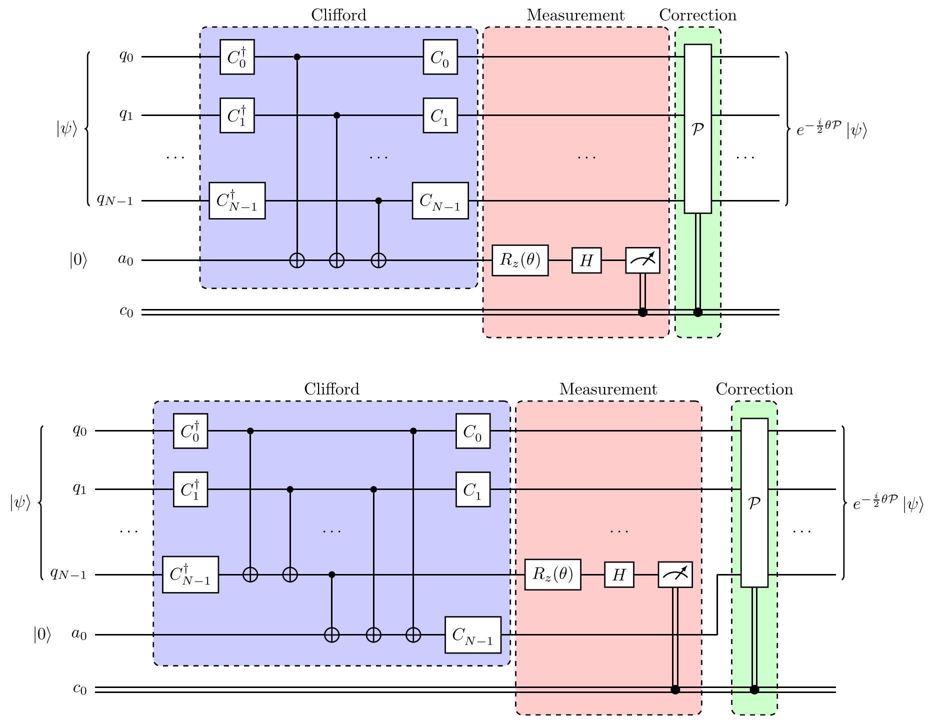


Figure 6. Patterns to implement $\exp(-\mathrm{i}/2\theta\mathcal{P})$ with the *star+ancilla* layout (top panel) and the *star* layout (bottom panel). Here, we assume that $\mathcal{P}$ involves no identities, thus, the local Clifford gates take the form $C_i \in \{H, SH\}$ and all main qubits are entangled with the ancilla register. The final correction is given by the Pauli string $\mathcal{P}$ itself. Note that in the bottom panel, the ancilla qubit a_0 takes the role of the last main qubit q_{N-1} after the measurement.

operations of the Pauli string (e.g. if the Pauli string inside the exponential is given by $X_1Z_3Y_6$, Pauli corrections are applied to qubits 1, 3 and 6 only). From figure 5 and equation (7), we can infer that the correction is precisely given by a controlled version of the Pauli string itself, since the Z corrections (implementing the diagonalized $\mathcal{Z}$ string) undergo a basis transformation according to the string. Schematically, this is shown in figure 6 for both layouts (*star* and *star+ancilla*).

At first glance, using the *star* layout has no benefits since more entangling gates are required. However, upon concatenation of multiple Pauli unitaries, the *star* gadget can lead to favourable gate cancellation properties. An example is later provided in section 5.2.1.

4.2. Parallel and adaptive measurements

To showcase an important property of our MB-circuit protocol, we now consider the case of two commuting Pauli strings Y_0Y_1 and X_0X_1, once again implemented through the *star+ancilla* layout. The naive circuit representation, which is obtained through concatenation of the circuits for $\exp(-\mathrm{i}/2\theta_2X_0X_1)$ and $\exp(-\mathrm{i}/2\theta_1Y_0Y_1)$, is depicted in the upper panel of figure 7.

Using the same circuit identities as before, we can shift the first correction and measurement layer across the second Clifford layer. In this particular case, this step introduces no corrections on the second ancilla qubit, which would have to be carried out before the second measurement. Instead, both measurements can be performed in parallel (see lower panel of figure 7). The fact, that the measurements can be parallelized here, is no coincidence. In the following, we derive the condition for parallelism.

Let us assume, we want to apply two unitaries generated by the Pauli strings $\mathcal{P}$ and $\tilde{\mathcal{P}}$. We now investigate the conditions, which these two Pauli strings have to fulfill, for parallel measurement. As explained before, the circuit implementing the matrix exponential of $\mathcal{P}$ ends with a controlled version of $\mathcal{P}$ itself, i.e. $\mathcal{P}^s$, where $s \in \{0,1\}$. Hence, if we want to bring the pattern implementing the product $\exp(-\mathrm{i}/2\tilde{\theta}\tilde{\mathcal{P}})\exp(-\mathrm{i}/2\theta\mathcal{P})$ into a standard form, it is sufficient to see what happens when shifting $\mathcal{P}^s$ through $\exp(-\mathrm{i}/2\tilde{\theta}\tilde{\mathcal{P}})$.

In appendix A we show that:

$$e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}}\mathcal{P}^s = \mathcal{P}^s \begin{cases} e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}}, & \text{if } \left[\mathcal{P},\tilde{\mathcal{P}}\right] = 0, \\ e^{-\frac{i}{2}(-1)^s\tilde{\theta}\tilde{\mathcal{P}}}, & \text{else,} \end{cases} \tag{9}$$

where s denotes the measurement outcome of the first ancilla qubit. Consequently, the product of two unitaries generated by two Pauli strings can be implemented in parallel only if the strings commute.

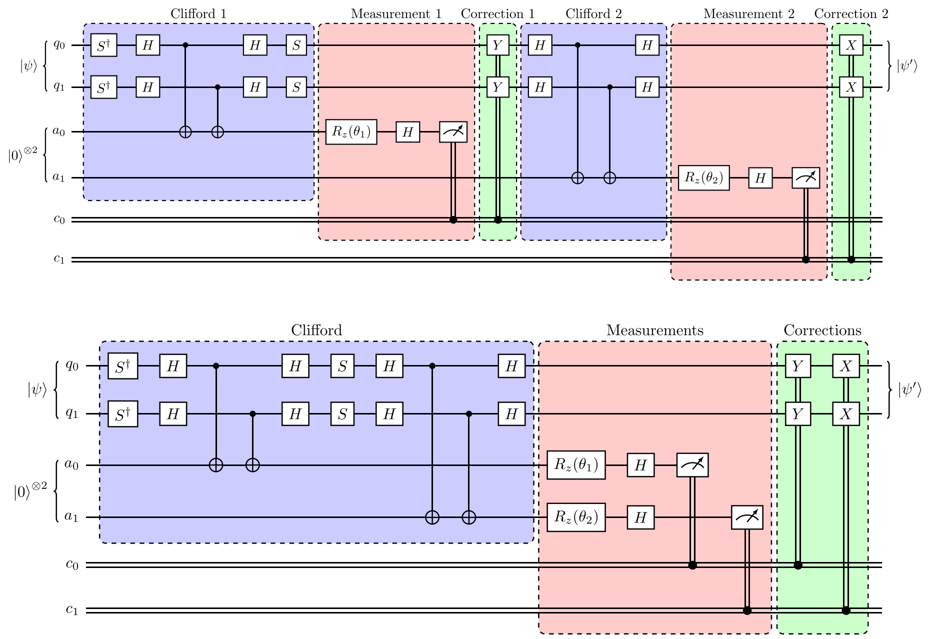


Figure 7. Upper panel: Naive circuit representation obtained through concatenation of the circuits for $\exp(-\mathrm{i}/2\theta_1 Y_0 Y_1)$ and $\exp(-\mathrm{i}/2\theta_2 X_0 X_1)$. Lower panel: Measurement-based pattern to implement $\exp[-\mathrm{i}/2(\theta_2 X_0 X_1 + \theta_1 Y_0 Y_1)]$ acting on the first two qubits. By rewriting the pattern into its standard form, the measurements of the two ancilla can be applied in parallel.

Otherwise, the rotation angle $\tilde{\theta}$ of the second Pauli exponential has to be adapted to the measurement outcome s of the first ancilla, leading to an adaptive, i.e. non-parallel measurement pattern.

Generalizing this result to the application of M unitatries generated by Pauli strings $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M\}$, we find that the final correction of the pattern implementing this operation in standard form is given by

$$\prod_{m=1}^{M} \mathcal{P}_m^{s_m}, \tag{10}$$

where s_m denotes the measurement outcome of the mth ancilla. Equation (10) allows to write down the final correction of an arbitrary measurement pattern, without the additional computational cost of propagating all corrections to the end of the circuit.

All ancilla qubits can be measured in parallel, only if all Pauli strings commute with each other. Otherwise, the measurement bases have to be adapted according to equation (9). In contrast to the final correction in equation (10), the adaptive measurement bases depend on the order in which the Pauli exponentials are implemented. To implement the unitary $\exp(-\mathrm{i}/2\theta_i \mathcal{P}_i)$, the rotation angle of the measurement basis is obtained by flipping the sign of θ_i for each previous non-commuting pattern measured in the $|1\rangle$-state, i.e.

$$\theta_i \to (-1)^{h_i} \theta_i, \quad \text{where} \quad h_i = \sum_{\substack{j<i \\ [\mathcal{P}_i, \mathcal{P}_j] \neq 0}} s_j. \tag{11}$$

To minimize the number of adaptive measurements, it is therefore convenient to first sort the generating strings into groups of mutually-commuting operators.

4.3. Applying quantum gates in constant depth in the circuit model

In this section, we show how our method can be used to derive measurement patterns, that can be used to apply several commuting operators in parallel with constant circuit depth in the circuit model. In the following, we assume that the depth of a quantum circuit is defined by the number of layers with at least one CNOT gate, which are needed to implement it.

Let us assume that we apply several commuting unitaries $U_1, \ldots, U_n$ generated by n Pauli strings to a quantum state $|\psi\rangle_N$—the order does not matter, since they commute. Then, we first derive the measurement pattern which implements $U = U_n \cdots U_2 \cdot U_1$ using the method outlined in the previous section. This pattern, in standard form, has three layers: Cliffords, measurements (in parallel) and corrections, cf figure 8.

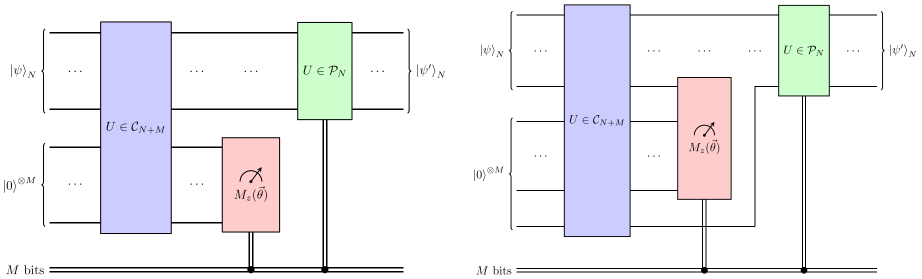


Figure 8. General circuit structure for M Pauli strings acting on N qubits using MBQC on arbitrary initial states with different architectures for the multi-qubit rotations. The left panel depicts the structure obtained through the *star+ancilla* layout (figure 3(a)). Here, all measurements are performed on the ancilla register. The right panel shows the circuit using the *star* layout (figure 3(b)), hence the first measurement occurs on the main register, thus teleporting the last main qubit through the entire ancilla register.

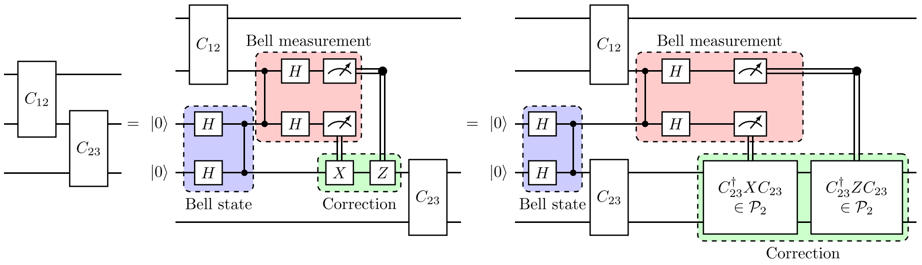


Figure 9. Parallelization of two-qubit Clifford gates by introducing ancilla qubits. By using quantum teleportation, any Clifford circuit can be reduced to constant depth.

Since the measurement and correction layer have already constant depth, we just need to implement the Clifford operations in constant depth.

For this, note that a general Clifford circuit can always be expressed as a sequence of one-qubit Cliffords and CNOT gates. Thus, it suffices to show that two sequential CNOT gates (with potentially intermediate one-qubit Cliffords) can be applied in parallel in constant depth. This can be achieved using the quantum teleportation algorithm. The general construction is depicted in figure 9: any sequence of two-qubit Clifford gates can be recast to a quantum process of constant depth. Using this technique, the number of ancilla qubits grows linearly with the depth of the Clifford layer, while the additional classical computation due to the corrections grows logarithmically [2].

To summarize, we can implement arbitrary Clifford circuits as constant depth circuits. Combining this with the previous result of deriving a pattern for a group of commuting operators (section 4.2), we conclude that our algorithm allows the implementation of several commuting operators as a constant-depth measurement-based pattern. More precisely, these constant-depth patterns can always be achieved with three entangling layers (one for the Bell state preparation of the ancilla qubits, one for the initial entangling gates and one for the Bell basis measurements), a measurement layer, and a corrective Pauli layer.

4.4. Simulation and correction of the main qubits

Up to this point, our observations hold for arbitrary input states. We now show how the circuits can be further reduced by simulating the Clifford part classically, assuming that the main qubits are initialized in a stabilizer state, i.e. the initial state can be prepared with Clifford gates acting on the $|0\rangle^{\otimes N}$ state.

We start by converting the Clifford part of the circuit into a graph state using the Clifford-simulation algorithm by Anders and Briegel [21] (the original code by Anders can be found online[5]), which we reimplemented using the `rustworkx` package [51]. The result of this conversion is a graph state of size $N_q + N_a$, where N_q denotes the number of initial qubits (main qubits) and N_a the number of required ancilla qubits, which is equivalent to the number of non-Clifford gates in the initial circuit. In this graph state, all main qubits are measured in a Pauli basis, while the ancilla qubits are measured in the $M(\theta)$ basis. Furthermore, after measuring the ancilla qubits, a final Pauli correction has to be applied to the main qubits.

[5] https://github.com/marcusps/GraphSim.

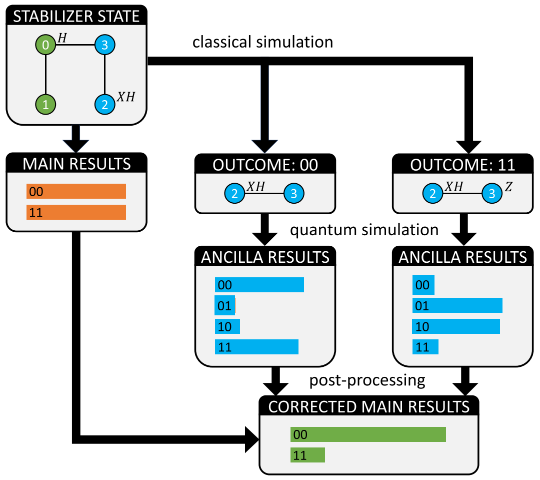


Figure 10. Overview how to simulate the main qubits. Starting with a graph state, in which the main qubits are measured in Pauli bases and the ancilla qubits in rotated bases, we can classically simulate the outcome of the main qubits. For each outcome, we derive the remaining state of the ancilla qubits before the measurement, which are all equivalent up to local unitaries.

In the MBQC protocol, we would now proceed by first measuring the ancilla qubits, correcting the main qubits depending on their outcome and then, at the end, perform measurements on the main qubits depending on the observables we wish to extract. However, since the main qubits are measured in the Pauli X, Y or Z basis, we can equally first simulate their measurement outcome and post-process the application of the Pauli corrections. This might seem surprising, but it can be exemplified as follows.

Suppose the main qubit q is corrected by Pauli-X depending on the measurement outcome s_a of an ancilla qubit a. Then, after the measurement of the ancilla, the new state of q is given by $X^{s_a}|q\rangle$ Let us assume that we now measure qubit q (in the Z-basis) *before* qubit a with probability p in $|1\rangle$ and $(1-p)$ in $|0\rangle$. Then we know that in the case of $s_a = 1$ we would have measured $|1\rangle$ with probability $(1-p)$ and $|0\rangle$ with probability p, if we had changed the order to first measuring qubit a and then q. Thus, we can equivalently flip the measurement outcome s_q of $|q\rangle$ depending on s_a. In the case of a Z correction nothing has to be done. For a Y correction we can always rewrite $Y = X \cdot Z$ up to an irrelevant, global phase.

Following this logic, we can first efficiently simulate the measurements of all main qubits using the graph state simulator, then execute the remaining circuits on the ancilla qubits and use a post-processing algorithm to correct the counts of the main qubits accordingly. More specifically, we first classically sample one shot on the main qubits neglecting the correction layer. Based on that, we obtain a bit string and a result-dependent stabilizer state (which is equivalent up to local unitaries to a graph state) for the ancilla qubits. This stabilizer state is then prepared and measured in the rotated bases on a quantum computer. If the ancilla measurement results imply an X or Y correction on the main qubits, the bit string is modified through the appropriate bit flips.

It is important to emphasize, that, while an exponential amount of classical measurement outcomes can occur, there is a one-to-one correspondence between a classically simulated shot in our algorithm and a quantum shot in the circuit model. Hence, the number of classical results that actually have to be considered is limited by the number of shots. Consequently, no exponential blowup occurs in this hybrid approach.

As an example, let us reconsider the measurement pattern that implements $\exp\left[-\mathrm{i}/2\theta(X_0X_1 + Y_0Y_1)\right]$ from figure 7 and let it act on the initial state $|00\rangle$. The overview over the full calculation is summarized in figure 10. After simulating the Clifford part of the circuit, we find the graph state shown in the top left. By using the Pauli measurement rules for graph states [46] integrated in the graph state simulator [21], we find two possible measurement-outcomes for the two main qubits: 00 and 11. Since we sample from a stabilizer state, all bit strings (with non-zero amplitudes) appear with equal probability [52]. All outcomes lead to local equivalent stabilizer states, where the ancilla qubits are decoupled from the main qubits. In our example, we find the two stabilizer states shown in the second row of figure 10. Running these stabilizer state on a quantum computer results in the probability distribution that is needed to correct the counts of the main qubits.

4.5. Optimization of graph states

The graph states obtained through our simulation protocol are often quite complicated. However, they are not unique. Stabilizer states are invariant under the operation of local complementations. In the following, we denote the local complementation of a graph at a vertex α by LC_α. Applying LC_α to a graph $G(V,E)$

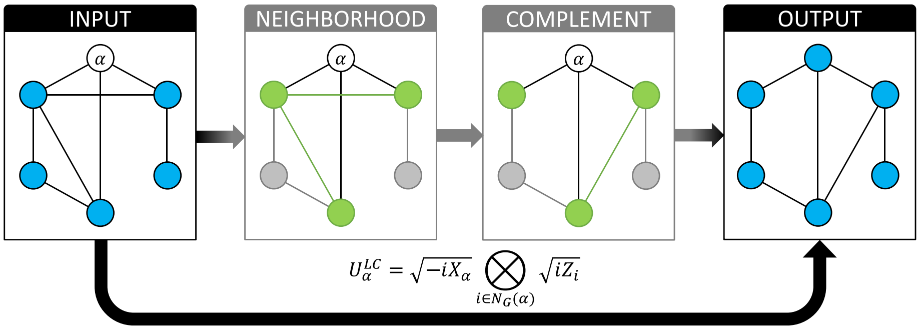

INPUT
NEIGHBORHOOD
COMPLEMENT
OUTPUT
α
$U_\alpha^{LC} = \sqrt{-iX_\alpha} \bigotimes_{i \in N_G(\alpha)} \sqrt{iZ_i}$

Figure 11. Local complementation of vertex α. In this example, the number of edges (i.e. the number of CZ gates to prepare the graph state) is reduced by one.

complements the neighborhood of the vertex α. That is, all existing edges between the vertices in $N_G(\alpha)$ are removed and all missing edges in $N_G(\alpha)$ are added. The corresponding stabilizer state is preserved by applying the local unitaries [21, 53]:

$$U_\alpha^{\text{LC}} = \sqrt{-iX_\alpha} \bigotimes_{i \in N_G(\alpha)} \sqrt{iZ_i}, \tag{12}$$

where $N_G(\alpha)$ denotes the neighborhood of the vertex α. The procedure is depicted in figure 11. The total number of edges may be changed after the operation.

With current NISQ-hardware in mind, we want to find optimal graph states with respect to the number of edges, which defines the number of CZ gates in the preparation circuit. Alternatively, we could optimize the states with respect to the maximum degree, thus minimizing the circuit depth required for preparation, or even optimize a trade-off between both properties.

To perform the optimization task, we employ the simulated annealing algorithm [23], which we will briefly outline here. A more detailed description is provided in appendix B. The solution space is the set of graph states that are LC-equivalent to the initial graph state obtained by converting the Clifford circuit. In each iteration, a random node of the graph is locally complemented. The cost function, we aim to minimize, is then evaluated with respect to the new graph state. If it is improved, the old graph state is discarded. Otherwise, we might still keep the new graph state, but only with steadily decreasing probability according to a Boltzmann distribution.

Despite its inherent simplicity and no guarantee to find the global optimum, we have observed major reductions in circuit depth results using this method.

5. Applications

In this section, we show how our methods can be applied directly to two important NISQ algorithms. In section 5.1 we show how combinatorial problems can be solved using the QAOA and in section 5.2 we show how the electronic-structure problem can be tackled by using the VQE. All simulations were performed using `qiskit` [54].

For the QAOA we derive an ansatz containing adaptive mid-circuit measurements due to non-commuting operators. For the VQE we design an ansatz which consists entirely of a set of commuting operators. In this scenario our method yields the most powerful reduction in circuit depth and does not rely on mid-circuit measurements, which are still challenging on current quantum computing devices.

5.1. QAOA

The QAOA [27, 28] is an optimization algorithm designed to solve combinatorial optimization problems in the NISQ-era [24]. The idea is to encode the optimization problem into a minimization problem of a generic Ising-Hamiltonian (also called cost Hamiltonian H_c)

$$H_c = \sum_{i<j} w_{ij} Z_i Z_j + \sum_i w_i Z_i, \tag{13}$$

where w_{ij} and w_i are coefficients depending on the optimization problem. More generally, the Hamiltonian can also include terms of higher locality, e. g. a 3-local optimization problem is given by

$$H_c = \sum_{i<j<k} w_{ijk} Z_i Z_j Z_k + \sum_{i<j} w_{ij} Z_i Z_j + \sum_i w_i Z_i. \tag{14}$$

These higher-order terms can be either reduced to two-local terms with the overhead of introducing additional qubits [55] or directly implemented on the hardware [56].

In its standard formulation, the QAOA algorithm tries to find the lowest energy (corresponding to the optimal solution of the initial problem) using a variational ansatz of depth p, which is given by

$$|\psi\rangle = e^{-i\beta_p/2H_m} e^{-i\gamma_p/2H_c} \cdots e^{-i\beta_1/2H_m} e^{-i\gamma_1/2H_c} |+\rangle^{\otimes N}, \tag{15}$$

where $H_m \equiv \sum_{i=1}^N X_i$ is the so-called mixer Hamiltonian, and N denotes the number of qubits. The γ_i and β_i are in total $2p$ variational parameter, which are obtained in a classical optimization feedback and aim to minimize the expectation value $\langle\psi|H_c|\psi\rangle$, which can be estimated efficiently on a quantum computer.

5.1.1. Circuit-based vs. Measurement-based QAOA

We test our graph-based compilation approach for generic QAOA instances with $p = 1$ and $p = 2$ for up to 4-local Hamiltonians with 40 qubits. We generate random instances and compare the cost of two-qubit gates as well as the required depth of the measurement-based vs. the circuit-based approach.

Since for QAOA the final measurement of the main qubits is always performed in the Z-basis, we classically pre-simulated this final measurement in the graphs to reduce their complexity. In order to further optimize the graph states inside the measurement patterns, we use simulated annealing as described in section 4.5. We use a cooling rate of 0.995 and temperatures of $T_{\text{initial}} = 100$ and $T_{\text{final}} = 1$. Each graph was optimized up to three times.

A comparison between the graph-based approach and the circuit model is shown in figure 12. Each data point in that figure corresponds to 10 different random instances, where the error-bars indicate the minimum and maximum and the dots or squares indicate the median of the shown quantities. In the upper panel we show the effect of optimizing the graphs with simulated annealing. Evidently, simulated annealing helps in most cases to reduce the numer of edges as well as to reduce the maximum vertex degree of the graphs. Only for the most difficult graphs (for $k = 4$, $p = 2$ and Hamiltonians with more than 70 terms) our optimization routine does not help to significantly reduce the complexity of the graphs.

In order to estimate the number of required two-qubit gates and the depth in the circuit model, we use the *star*-layout to implement the circuits. In this layout, the number of CNOT gates required to implement n Pauli unitaries with Pauli string of locality k is given by

$$2 \cdot (k-1) \cdot n. \tag{16}$$

In order to optimize the depth of the circuits and perform as many two-qubits gates as possible in parallel, we use a greedy approach where we search in a given layer as many unitaries that can be applied in parallel as possible.

In the lower panel of figure 12 we compare the number of CNOTs and the required depth of the circuit model vs. the graph-based approach. Gate- and depth ratios are calculated as the required number of CNOT gates and depth of the circuit divided by the number of edges of the graph and depth of the measurement pattern. Note, that the depth of the pattern includes the depth required to prepare the graph state and the number of required adaptive measurements. As can be seen in the plot, for two-local Hamiltonians the circuit-based QAOA outperforms our graph-based approach in the number of required two-qubit gates. For the required depth, we find a small improvement for small Hamiltonians, but with increasing number of terms in the Hamiltonian, the circuit-based approach also results in shorter depths. For three- and four-local Hamiltonians the situation is a bit different. While the graph-based approach still is outperformed for $p = 2$ in the number of required two-qubit gates by the circuit model, the measurement patterns seem to have an advantage in the required depth. However, for an increasing number of terms in the Hamiltonian this advantage gets smaller such that for real-world applications the circuit model seems always to be the better option. The reason for this could be that the simulated annealing approach to minimize the edges of the graphs does not fully reduce them due to their complexity. More sophisticated optimization problems might help here.

Our approach to first create the complete graph state on which the measurements are performed to implement QAOA could still be beneficial in some situations. The advantage of using graph states lies in the fact that it can be easily embedded in bigger graphs, which can be tailored to specific hardware architectures

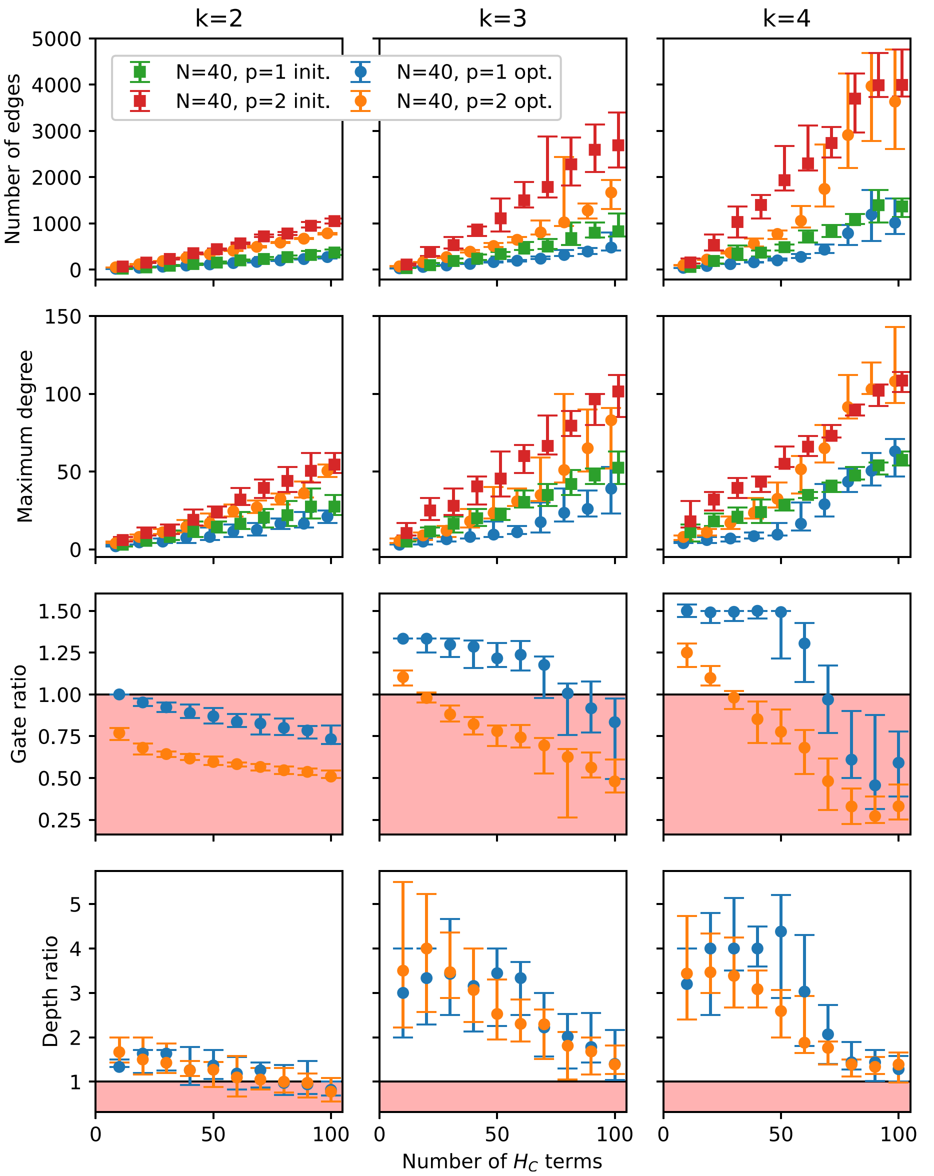


Figure 12. Upper panel: Scaling of the graphs to implement QAOA with ansatz depths of $p = 1$ and $p = 2$ for 40 qubits and sizes up to 100 terms in the Hamiltonian before and after optimization. We show data for Hamiltonians with different localities ranging from $k = 2$ to $k = 4$. The error-bars indicate the minimum and maximum values after running 10 random instances and the points indicate the median. Lower panel: Comparison of required CNOT gates and depth of the graph-based method vs. the circuit implementing the same ansatz.

and which also allows the reduction of the maximum degree of the graph. In principle, any graph state with n qubits and m edges can be prepared in constant depth using $\mathcal{O}(m)$ ancillary qubits, as has been shown by Høyer *et al* [47]. Thus, by reducing the number of edges in the graphs using simulated annealing, the optimized graphs can be parallelized more efficiently using fewer quibts.

5.1.2. Hardware experiment

As a concrete example and a proof-of-principle of our method, we consider a weighted max-cut problem with four vertices. The task of the weighted max-cut problem is to find a partitioning of the vertices in two complementary sets, such that the sum of all weights on the cut is maximized. In figure 13 we show the graph and the optimal partitioning, which is given by dividing the vertices into the two sets $\{0,1,3\}$ and $\{2\}$ in our example. The weighted max-cut problem can be formulated as a minimization problem of an Ising-Hamiltonian. In our case the Hamiltonian is given by

$$H_\mathrm{c} = Z_2 Z_3 + Z_0 Z_2 + 0.5\, Z_0 Z_1 + Z_1 Z_2. \qquad (17)$$

The optimal solution is given by the two bit strings (here and in the following we use the little-endian convention) 0100 and 1011, corresponding to the two sets mentioned above.

Table 1. Local unitaries for the pattern implementing the QAOA ansatz.

Counts of main qubits	Local unitaries $\{C_0, \dots, C_7\}$
$\{0, 15\}$	$\{I, Y, YH, Z, I, H, I, Z\}$
$\{8, 7\}$	$\{I, Y, YH, Z, Z, H, I, Z\}$
$\{4, 11\}$	$\{I, Y, YH, Z, Z, H, Z, I\}$
$\{12, 3\}$	$\{I, Y, YH, Z, I, H, Z, I\}$
$\{2, 13\}$	$\{I, Y, YH, Z, I, H, Z, Z\}$
$\{10, 5\}$	$\{I, Y, YH, Z, Z, H, Z, Z\}$
$\{6, 9\}$	$\{I, Y, YH, Z, Z, H, I, I\}$
$\{14, 1\}$	$\{I, Y, YH, Z, I, H, I, I\}$

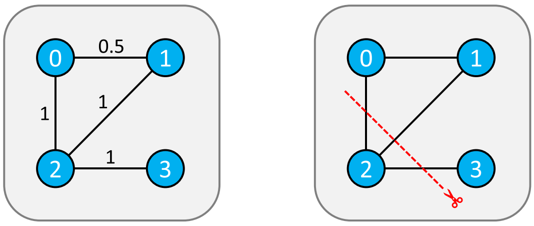


Figure 13. Example of the weighted max-cut problem. Given a weighted graph (left), the task is to find a partitioning which maximizes the sum of all weights on the cut (right). In this example, the best cut goes through all edges with weight one, partitioning the graph into the nodes $\{0, 1, 3\}$ and $\{2\}$.

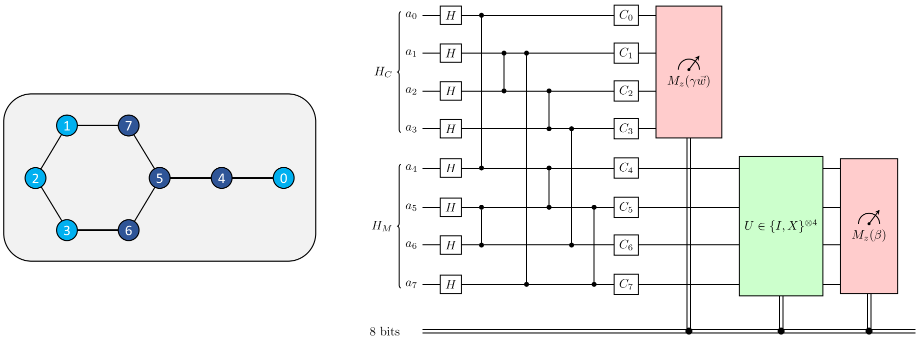


Figure 14. Left panel: graph state that implements the QAOA ansatz. The qubits $q_0 - q_3$ correspond to the unitary implementing $\exp(-\mathrm{i}\gamma/2H_\mathrm{c})$, while qubits $q_4 - q_6$ implement $\exp(-i\beta/2H_m)$. Right panel: circuit which implements the graph state, together with the local unitaries which depend on the measurement outcome from the simulation of the main qubits, and the adaptive measurement instructions.

To solve the problem with QAOA we use the $p = 1$ ansatz, which has two parameters:

$$|\psi\rangle = e^{-i\frac{\beta}{2}(X_0+X_1+X_2+X_3)} e^{-i\frac{\gamma}{2}H_\mathrm{c}} |+\rangle^{\otimes N}. \tag{18}$$

From classical simulation we find the optimal angles $\gamma = -2.290$ and $\beta = -2.186$.

We first derive the pattern which implements this unitary, by mapping the two operators $e^{-i\frac{\gamma}{2}H_\mathrm{c}}$ and $e^{-i\frac{\beta}{2}(X_0+X_1+X_2+X_3)}$ to stabilizer states. Since both operations do not commute, this introduces adaptive measurements. We then simulate the main qubits and identify the graph shown in the left panel of figure 14 as the full pattern, that implements the variational QAOA ansatz. Depending on the pre-simulated counts of the main qubits, we show the additional local unitaries in table 1. The counts are encoded using the the little-endian convention, e.g. the count $8 = (1000)_2$ corresponds to the case, in which the third main qubit was measured as 1 and the other three as 0.

We ran the circuit corresponding to this pattern (right panel of figure 14) on the 27-qubit quantum computer `ibm_auckland`, using dynamic circuits. The QAOA distribution, which is obtained by correcting the pre-simulated counts of the main qubits depending on the counts of the ancilla circuits, is shown in figure 15. We compare the results from `ibm_auckland` with ideal results from simulation. As expected, we find with highest probability the correct bit string 0100 and 1011 in the simulated as well as in the measured distributions. However, due to hardware noise, the measured distribution differs from the ideal one.

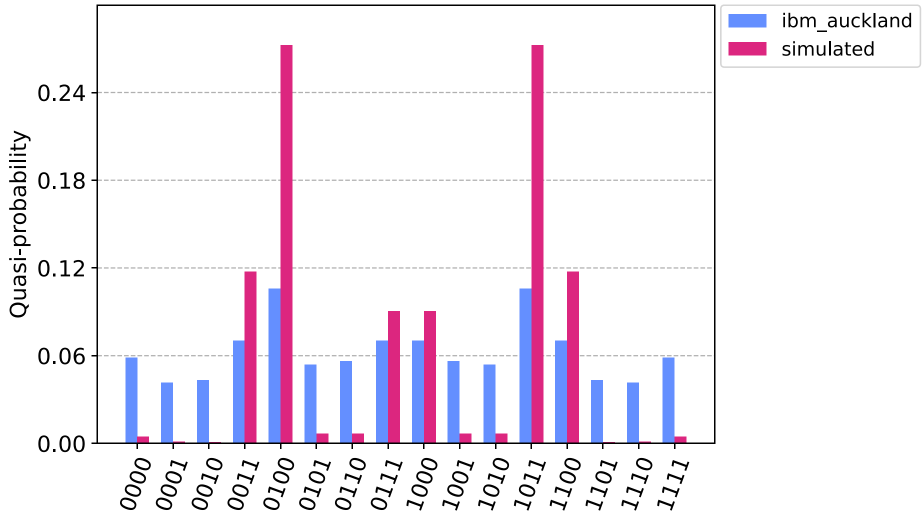


Figure 15. QAOA counts on `ibm_auckland` using dynamic circuits and ideal, simulated counts. As expected, in the ideal distribution the correct solutions 0100 and 1011 are measured most often. The normalized fidelity (cf equation (19)) between the ideal and simulated distribution is $F = 0.28$.

To quantify the error, we calculate the normalized fidelity between the two distributions [57]

$$F\left(P_{\text{ideal}}, P_{\text{measured}}\right) = \frac{F_{\text{H}}\left(P_{\text{ideal}}, P_{\text{measured}}\right) - F_{\text{H}}\left(P_{\text{ideal}}, P_{\text{depol}}\right)}{1 - F_{\text{H}}\left(P_{\text{ideal}}, P_{\text{depol}}\right)}, \tag{19}$$

where F_{H} denotes the Hellinger fidelity and P_{depol} corresponds to a uniform distribution, which would be measured on a completely depolarized device. In our experiment, we find a fidelity of only $F = 0.28$.

We believe, that the main source of error is due to the use of dynamic circuits and measurement errors, which affect the whole outcome, if they occur during a mid-circuit measurement. Dynamic circuits are a fairly new feature on IBM quantum computers. For the next application, the VQE, we therefore construct circuits, in which we avoid mid-circuit measurements by designing an ansatz with commuting operators only.

5.2. VQE for quantum chemistry

Quantum chemistry is often discussed as one of the most promising fields in which quantum computing could have a big impact. An important task in quantum chemistry is the determination of the molecular ground-state energy, which is given by the minimum of the molecular Hamiltonian. In second quantization, it is typically expressed in terms of fermionic annihilation and excitation operators,

$$H = \sum_{p,q} h_{pq} a_p^\dagger a_q + \sum_{p,q,r,s} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s, \tag{20}$$

where a_p ($a_p^\dagger$) annihilates (creates) an electron in the spin–orbital p and the ground-state energy corresponds to the minimal eigenvalue. To find the minimal eigenvalue of equation (20) using quantum computers, the VQE has been thoroughly studied in the past decade [25, 26, 58–61].

The unitary-coupled cluster (UCC) ansatz is among the most popular VQE ansätze and is defined as

$$|\psi\rangle = e^{\sum_n T_n} |\Phi\rangle_0, \tag{21}$$

where $|\Phi\rangle_0$ is the reference state (usually the Hartree–Fock ground state) and T_n denote the nth cluster operator—usually these are truncated at second order, where T_1 and T_2 are given by

$$\begin{aligned} T_1 &= \sum_{\substack{i \in \text{virt.} \\ a \in \text{occ.}}} \theta_a^i \left(a_i^\dagger a_a - a_a a_i^\dagger\right), \\ T_2 &= \sum_{\substack{i,j \in \text{virt.} \\ a,b \in \text{occ.}}} \theta_{ab}^{ij} \left(a_i^\dagger a_j^\dagger a_a a_b - a_b^\dagger a_a^\dagger a_j a_i\right), \end{aligned} \tag{22}$$

where virt. (occ.) denotes the set of virtual (occupied) orbitals. Using the Jordan–Wigner mapping, those operators can be mapped to qubit operators via

$$a_n^\dagger \to Z_0 Z_1 \cdots Z_{n-1} \frac{X_n + i Y_n}{2}. \tag{23}$$

For the double-excitations T_2, this substitution leads to a sum of eight Pauli-strings. In the literature these operators are often simplified by neglecting all Z_i terms. The operators in this approximation are called qubit excitation operators [59, 60]. For instance, the qubit excitation corresponding to a fermionic double excitation operator in the Jordan–Wigner mapping is given by:

$$U_{ijab} = e^{i\frac{\theta}{8} X_i Y_j X_a X_b} e^{i\frac{\theta}{8} Y_i X_j X_a X_b} e^{i\frac{\theta}{8} Y_i Y_j Y_a X_b} e^{i\frac{\theta}{8} Y_i Y_j X_a Y_b} e^{-i\frac{\theta}{8} X_i X_j Y_a X_b} e^{-i\frac{\theta}{8} X_i X_j X_a Y_b} e^{-i\frac{\theta}{8} Y_i X_j Y_a Y_b} e^{-i\frac{\theta}{8} X_i Y_j Y_a Y_b}. \tag{24}$$

5.2.1. Measurement pattern for double excitations

Next, we showcase two ways how the qubit double-excitation in equation (24) could be implemented in our protocol as a measurement pattern based on the standard forms provided in figure 8. We emphasize that these patterns hold for arbitrary input states, i.e. also non-stabilizer states. In the following, for the sake of convenience, we rescale $\theta \to 4\theta$ such that all R_z-gates perform rotations by an angle of $\pm\theta$. Note that the actual measurements here are constant-depth since the generators of the double-excitation operator mutually commute. According to equation (10), the final Pauli correction is given by

$$\begin{aligned} U = & \left(X_i X_j X_a Y_b\right)^{s_1} \cdot \left(X_i X_j Y_a X_b\right)^{s_2} \cdot \left(X_i Y_j Y_a Y_b\right)^{s_3} \cdot \left(Y_i Y_j Y_a X_b\right)^{s_4} \\ & \cdot \left(Y_i X_j Y_a Y_b\right)^{s_5} \cdot \left(Y_i X_j X_a X_b\right)^{s_6} \cdot \left(Y_i Y_j X_a Y_b\right)^{s_7} \cdot \left(X_i Y_j X_a X_b\right)^{s_8} \end{aligned} \tag{25}$$

for both patterns.

This qubit-excitation operator has also been considered in [61], in which its circuit has been derived and optimized using the *star* layout. In its optimized form, the circuit entails 13 CNOT gates with a depth of 13 and is shown in figure 16(a). We use the same gate cancellation techniques as Nam *et al* [61] to derive the measurement pattern with the *star* layout, as presented in figure 16(b). Using this procedure, we obtain a Clifford circuit which consists of 21 CNOT gates and has a depth of 12, which we calculated using `qiskit` [54]. The Clifford circuit constitutes of two portions with the first one essentially entailing the same 13 CNOT gates as in the circuit model, and the second one being a CNOT-ladder taking care of the gate-teleporation.

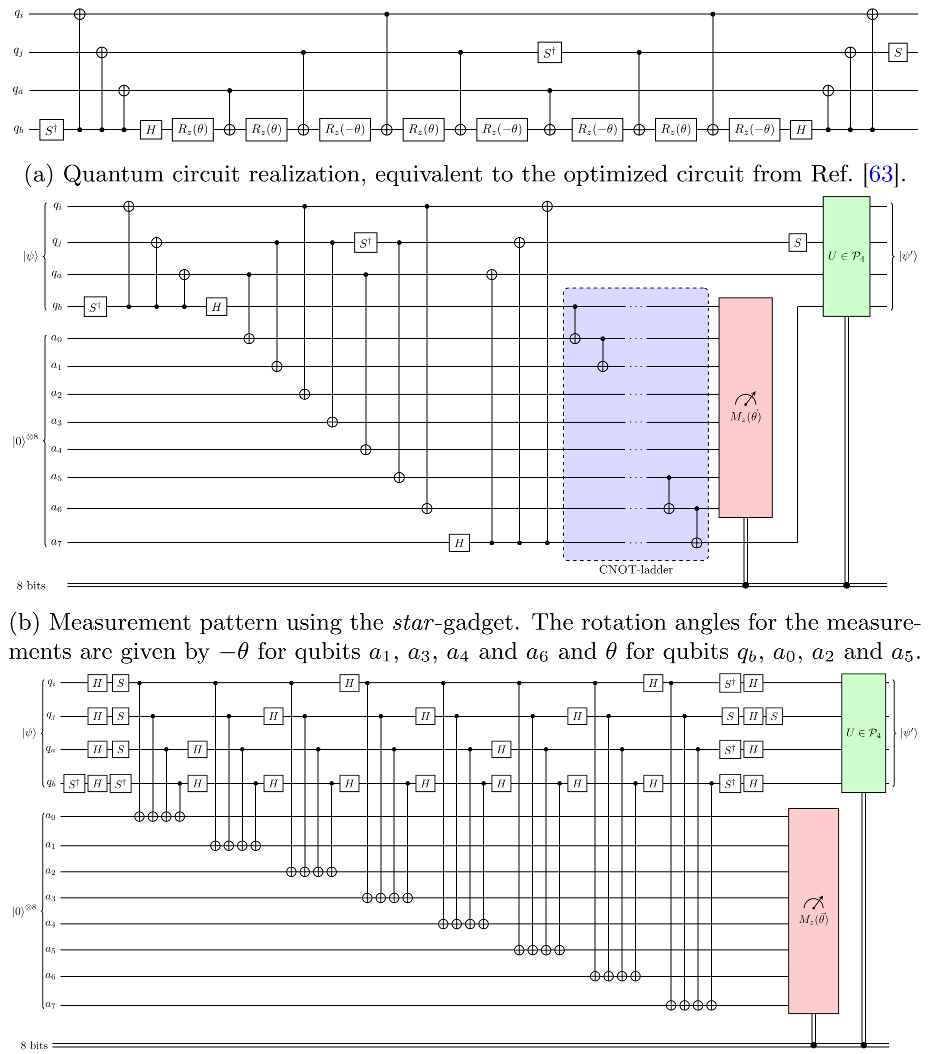
In figure 16(c), we provide an equivalent pattern based on the *star+ancilla* layout. We obtain a sequence of 32 CNOT gates which cannot be trivially cancelled. However, they can be arranged in a depth of 11. By adding the depth-one measurements, we find that our *star* pattern has a total depth of 13, which is the same as the optimized circuit from [61]. For the *star+ancilla* case, we achieve a total depth of 12, thus outperforming the optimized circuit from [61].

As already discussed in section 4.3, the depth of the Clifford circuits could be further reduced to constant depth circuits. For that, one needs to introduce two ancilla qubits everytime a CNOT gate shares a qubit with a previous CNOT gate. For the *star* layout, that would mean additional 2 × 30 qubits, and for the *star+ancilla* layout 2 × 52. Each pair of ancilla qubits is accompanied by two *CZ* (or CNOT) gates for the Bell-state preparation and Bell measurement, making a constant-depth implementation of the *star+ancilla* pattern much more expensive both in terms of qubits and gates. Finally, a detailed overview of the implementation costs of the different approaches is shown in table 2. This example proves that a sequence of Pauli unitaries might be more efficiently implemented with the *star* layout, showcasing that the *star* pattern can be beneficial compared to the already established *star+ancilla* pattern/phase gadget.

5.2.2. Circuit-based vs. Measurement-based VQE

We test our approach of mapping VQE ansätze to graph states for qubit-ADAPT [29]. In qubit-ADAPT, the qubit double excitation operators are simplified to individual Pauli strings consisting of three Pauli X and one Pauli Y. Compared to the qubit UCC ansatz with operators built as in equation (24), this approach considerably reduces the depth of the ansatz and leads to much simpler circuits.

To test the performance of a graph-based qubit-ADAPT, we generate generic ansätze for three different system sizes ($N = 20$, 40 and 60 qubits, where we assume half filling, i. e., 10, 20 or 30 electrons distributed in 20, 40 and 60 spin–orbitals) and generate random Pauli strings of the form $X_i Y_j X_a X_b$ or $X_i X_j Y_a X_b$ where $i,j < N/2$ and $a,b \geqslant N/2$. From these Pauli strings we generate graph states using our conversion algorithm. As for QAOA, we then simplify these graphs using simulated annealing with a cooling rate of 0.995 and temperatures of $T_\text{initial} = 100$ and $T_\text{final} = 1$.



(a) Quantum circuit realization, equivalent to the optimized circuit from Ref. [63].

(b) Measurement pattern using the *star*-gadget. The rotation angles for the measurements are given by $-\theta$ for qubits a_1, a_3, a_4 and a_6 and θ for qubits q_b, a_0, a_2 and a_5.

(c) Measurement pattern using the *star+ancilla*-gadget. The rotation angles for the measurements are given by $-\theta$ for qubits a_0, a_1, a_2 and a_4 and θ for qubits a_3, a_5, a_6 and a_7.

Figure 16. Different realizations of the double-excitation operator U_{ijab} from equation (24) using (a) the quantum circuit model. Reproduced from [61]. CC BY 4.0., (b) the *star*-gadget, and (c) the *star+ancilla*-gadget.

Table 2. Comparison of the implementation cost for the patterns shown in figure 16. The two columns on the right show the cost of the constant-depth patterns obtained by applying the teleportation trick to the patterns in figure 16. For the pattern depth, the notation $A+B$ means a CNOT-depth of A and B sequential measurements.

				For const. depth:	
	Ancillas	CNOTs	Pattern depth	Ancillas	CNOTs
star	8	23	$12+1$	$8+60$	83
star+ancilla	8	32	$11+1$	$8+104$	136

Note that, in contrast to the QAOA considered in section 5.2.2, we do not classically pre-simulate the measurement of the main qubits here. However, for a specific molecule, this can be done by grouping the Hamiltonian into commuting terms and designing Clifford circuits to simultaneously measure these groups [20], as we will see in the next section for H_2O. These Clifford circuits can then be appended to the pattern and—after standardization—absorbed into the graph state.

In figure 17 we compare for different ansatz sizes the number of required two-qubit gates as well as the depth of the graph-based vs. the circuit-based approach. As before, each data point corresponds to 10 different random instances and the error-bars indicate the minimum and maximum, while the dots or squares indicate the median of the shown quantities.

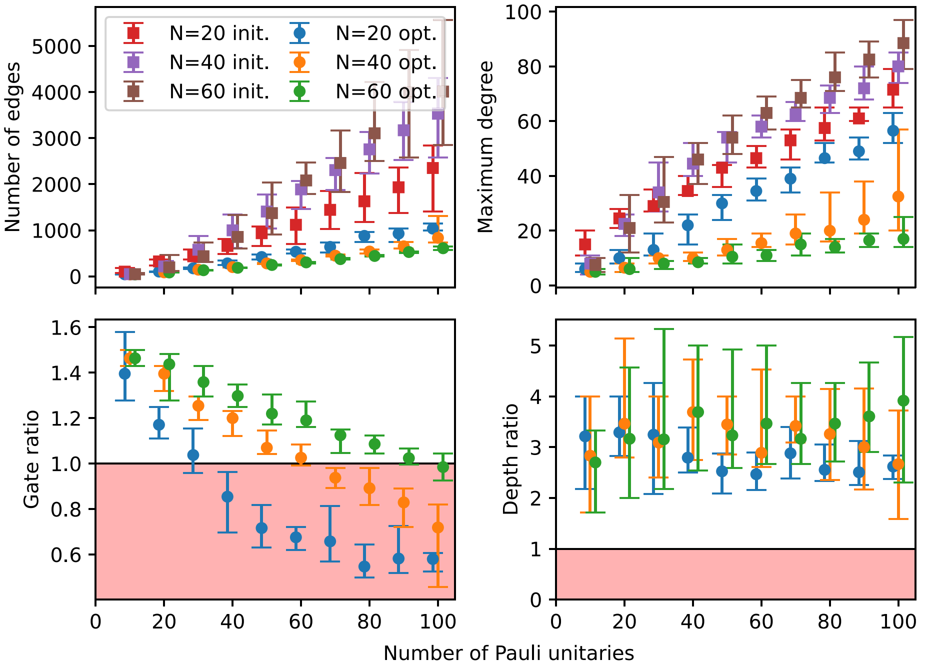


Figure 17. Upper panel: Scaling of the graphs to implementing quibt-ADAPT with ansatz sizes up to 100 operators. We compare the inital graphs with the optimized ones after simulated annealing—a drastic reduction in the number of edges as well as in the maximum degree of the graphs can be achieved by this optimization. Lower panel: Comparison of required CNOT gates and depth of the graph-based ansatz vs. the circuit implementing the same ansatz. The error-bars indicate the minimum and maximum values after running 10 random instances and the points indicate the median.

As can be seen in the upper panel of the figure, the simulated-annealing based optimization approach again drastically reduces the number of edges as well as the maximum degree of the different graph states, leading to optimized graph states that can be prepared in a reduced depth.

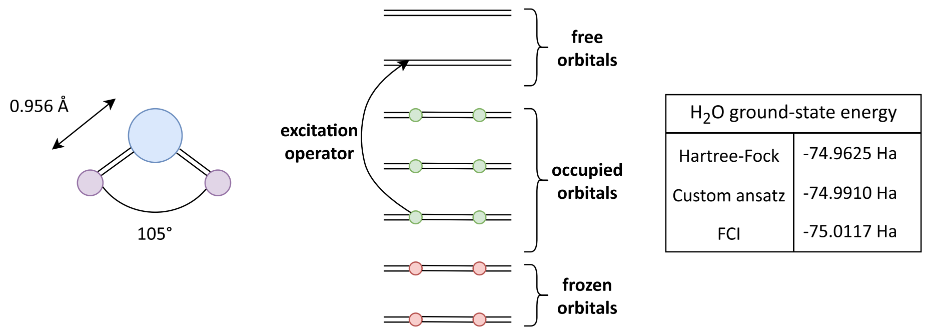
In the lower panel we show gate- and depth ratios, which are defined in the same way as in the case of QAOA (required CNOT gate and depth in the circuit model divided by required edges in the graph and depth of the pattern). We estimate the required number of CNOT gates and the depth in the circuit model in the same way as for the QAOA circuits by employing a greedy search for maximal parallel layers. From the gate-ratio plot we see that the graph-based approach can have advantages for smaller ansätze. However, for all system sizes ($N = 20$, 40 and 60 qubits), there is a cross-over point, where the circuit-based model needs less two-qubit gates than the measurement pattern. For the depth comparison the situation is different. Here we find an improvements of a factor 2 to 5 for all instances and all system sizes. This shows that it can be beneficial to map the qubit-ADAPT ansatz to a graph, since the parallel preparation of the graph state can lead to shorter circuits—at the expense of additional qubits. We want to emphasize that the benefit of using our graph-based approach becomes even more prominent when appending circuits for simultaneous measurements of some complex Hamiltonian such as for electronic-structure problems.

5.2.3. Hardware experiment: Ground-state energy estimation of the H_2O molecule

In this section we use our techniques to lower the quantum-resource requirements of the VQE to estimate the ground-state energy of the H_2O molecule on a quantum computer. Our starting point is the electronic-structure Hamiltonian in the minimal `sto-3 g` basis, which we derive using `openfermion` [62] together with `pyscf` [63]. The full Hamiltonian consists of 14 spin–orbitals, which are occupied by ten electrons. In order to simplify the problem at hand, we freeze the four spin–orbitals with the lowest energy, such that we only deal with six electrons distributed over ten orbitals in our ansatz. A schematic overview of this approach is provided in figure 18.

Next, we choose our variational ansatz for the VQE. The full UCC-ansatz in this case would result into too complicated patterns for current quantum hardware. Therefore, as in the previous section, we use the qubit-ADAPT ansatz to find the ground-state of the molecule. As shown in section 4.2, we can parallelize all operations, if they commute. We therefore aim to build our operator pool from commuting operators only. The full pool, consisting of nine operators built from Pauli strings, is shown in table 3.

Our ansatz is inspired by the ansatz from Nam *et al* [61], in which the most important qubit excitations were chosen—instead of using the full qubit excitation operators consisting of eight Pauli strings each, we



H_2O ground-state energy	
Hartree-Fock	-74.9625 Ha
Custom ansatz	-74.9910 Ha
FCI	-75.0117 Ha

Figure 18. Left panel: Molecular geometry of the H_2O molecule. Middle panel: Molecular orbitals contributing to our ansatz using the `sto-3 g` basis. We freeze the four electrons occupying the orbitals with the lowest energy and consider only excitations for the remaining six. Right panel: Ground-state energy of the H_2O in the Hartree–Fock approximation, using our variational ansatz with frozen electrons (with optimized parameters) and the FCI energy, which is obtained by estimating the lowest eigenvalue of the full Hamiltonian.

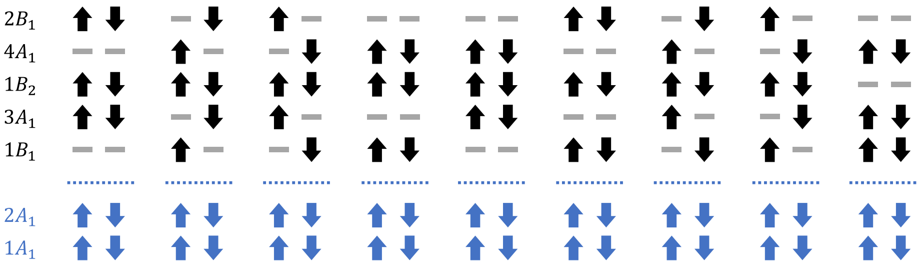


Figure 19. The nine most important excitations contributing to the ground-state energy of the H_2O molecule following [61], in the approximation of the four inner-most electrons being frozen. Reproduced from [61]. CC BY 4.0.

Table 3. Operator-pool for the VQE to estimate the ground-state of the H_2O molecule. We construct nine commuting Pauli operators to avoid mid-circuit measurements.

Operator P_n	Parameter θ_n
$X_0X_1Y_8X_9$	−0.157
$X_2X_3Y_8X_9$	−0.080
$X_4X_5Y_8X_9$	−0.023
$X_0X_1Y_6X_7$	−0.078
$X_2X_3Y_6X_7$	−0.081
$X_4X_5Y_6X_7$	−0.054
$X_1X_2Y_6X_9$	0.099
$X_0X_3Y_6X_9$	−0.067
$X_1X_2X_7Y_8$	−0.065

only choose one string per excitation. Following [61], the most important excitations using the frozen core approximation, are shown in figure 19. Our ansatz is then given by:

$$|\psi\rangle = e^{-\frac{i}{2}\sum_n \theta_n P_n}|\Phi\rangle_{\mathrm{HF}}, \tag{26}$$

where $|\Phi\rangle_{\mathrm{HF}} \equiv |0000111111\rangle$ is the Hartree–Fock ground state and θ_n denotes the nth variational parameter.

We first optimize our ansatz by performing a classical simulation of the circuit. The optimal parameters leading to a minimal energy of −74.9910 Ha are shown in table 3. In figure 20 we show the quantum circuit corresponding to this ansatz.

The Hamiltonian in our approximation consists of 251 terms in total. As observed in [19, 20], the sampling overhead of measuring the expectation value of such Hamiltonian can be reduced significantly by measuring commuting operators simultaneously. Using the `qiskit` command `group_commuting()` to identify commuting groups of a given operator, we find a partitioning of the Hamiltonian into 14 groups of operators, which can be measured simultaneously. More details to this decomposition can be found in

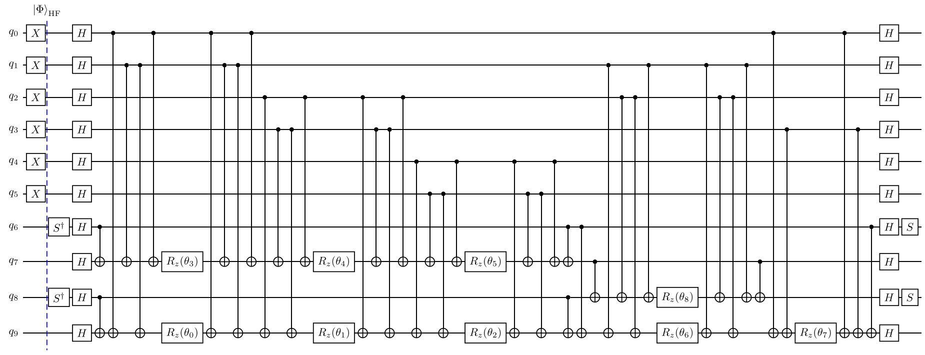


Figure 20. VQE ansatz in the circuit model. To measure the expectation of one of the 14 Hamiltonians from equation (27), a specific Clifford circuit must be appended at the end [19, 20]. This leads to high-depth circuits, which we reduce to shallow-depth pattern by mapping the ansatz to graph states.

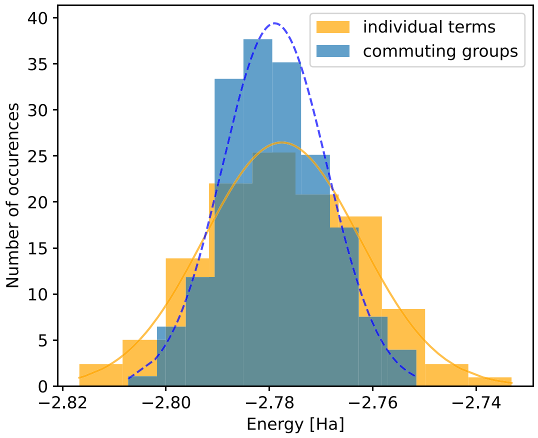


Figure 21. Comparison between the two sampling strategies. The data was obtained by simulating the quantum circuit shown in figure 20(with optimized parameters) in `qiskit` 500 times. For the case of measuring each term of the Hamiltonian individually, we use a total amount of roughly 250*k* shots in the simulation, while for the other approach we used only 70*k*. There is an advantage by measuring commuting groups simultaneously compared to measuring each term individually, since the standard deviation due to shot noise is smaller in this approach.

appendix C. Our full Hamiltonian reads:

$$H = -72.2129\ \mathrm{Ha} + \sum_{n=1}^{14} H_n. \tag{27}$$

In order to measure the expectation value of each Hamiltonian H_n, we use the techniques outlined in [20] to derive the Clifford circuits, which diagonalize a given set of commuting operators.

To showcase the reduction of sampling overhead in this approach, we simulate the quantum circuit shown in figure 20 using the `qiskit` simulator. In figure 21 we compare the two strategies of estimating the expectation value of the Hamiltonian measuring each term individually vs. using a partitioning into commuting groups. In the first approach we measure each term of the Hamiltonian with 1000 shots individually and in the second we use the technique of measuring each of the 14 groups with 5000 shots per group simultaneously. We repeated the simulation 500 times to collect sufficiently statistics to perform a fit with a normal distribution to estimate the variance of the expectation value due to shot noise for both strategies. While in the first approach we obtain a standard deviation of $\sigma \approx 0.015$ using approximately 250*k* shots in total, the second approach leads to results with less shot noise ($\sigma \approx 0.01$) using only 70*k* shots. This shows that using commuting groups of operators indeed gives a benefit for the measurement of the Hamiltonian in this example.

In the following, we construct the variational quantum circuit for each of the 14 groups by concatenating the ansatz circuit (figure 20) with the corresponding Clifford circuit, that diagonalizes all operators in a given group (appendix C). We then use our algorithm to derive measurement patterns, which implement the whole quantum operation. After classically simulating the main qubits, we end up with very simple graph states after using the optimization procedure outlined in section 4.5. While the ansatz circuit figure 20

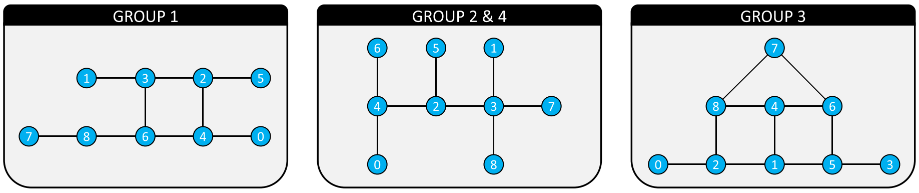


Figure 22. Graph states corresponding to the measurement of the first four terms of the Hamiltonian (27). Each graph represents 32 locally-equivalent states contributing to the expectation value.

Table 4. Expectation values of the Hamiltonians (and their respective standard deviations) measured on the quantum computer in comparison with the ideal ones. As a proof-of-principle we only measured the first four Hamiltonians.

Hamiltonian	Exp. val. ideal	Exp. val. measured	Exp. val. mitigated
H_1	-0.2565	-0.4396 ± 0.0014	-0.2698 ± 0.0024
H_2	-0.2546	-0.1852 ± 0.0008	-0.2460 ± 0.0008
H_3	1.0223	0.7691 ± 0.0026	1.0143 ± 0.0028
H_4	-0.1223	-0.0924 ± 0.0007	-0.1223 ± 0.0008

contains in total 44 CNOT gates (and a depth of 24), we find that the optimized patterns contain only up to 14 (with a maximum depth of 5), reducing the number of CNOT gates by a factor of at least three and the depth by a factor of at least four. This reduction becomes even more drastic, if we consider the simultaneous measurement of mutually-commuting groups in the circuit model. The number of CNOT gates increases due to the additional Clifford circuits at the end.

Next, we benchmark our ansatz using the optimized parameters and ran the corresponding circuits on the 27-qubit quantum computer `ibm_hanoi`. As a proof-of-principle, we only ran those circuits associated with the energies of the first four terms of the Hamiltonian. We show the graph states corresponding to these four terms in figure 22. We report our final results in table 4. For each group, we find that 32 LC-equivalent stabilizer states have to be prepared. Accordingly, we ran 32 circuits per group using $4k$ shots, such that each expectation value was measured with a total budget of $128k$ shots. The error for the raw data shown in table 4 (third column) are estimated by repeating each experiment eight times and calculating the standard deviation.

As can be seen from table 4, the energies calculated with the raw data points are far-off the ideal values, such that the use of error-mitigation is imperative. We mitigate read-out error using the `M3` package [64], for which we ran all calibration circuits with $100k$ shots. Additionally to the read-out error correction we used randomized compiling [65], dynamical decoupling [66] and zero-noise extrapolation (ZNE) [67, 68] through local gate folding [68] to increase the precision. The mitigated values are shown in the last column of table 4. The reported error follows from the ZNE fit parameter.

It is important to note that the applicability of ZNE is not evident here, as observables are measured classically on the main qubits and only later corrected through the ancilla outcomes. In appendix F we prove that the expectation value of an arbitrary Pauli string $\mathcal{P}_q$, acting purely on the main qubits, can be expressed as a superposition of expectation values of an auxiliary Pauli-Z string $\mathcal{Z}_a$ acting on the ancilla register, that is

$$\langle \mathcal{P}_q \rangle = \frac{1}{N} \sum_n (-1)^{s_n} {}_n\langle \psi_a | \mathcal{Z}_a | \psi_a \rangle_n, \tag{28}$$

where N is the number of distinct classical measurement outcomes, $|\psi_a\rangle_n$ are the ancilla states corresponding to the classical measurement outcome n and $s_n \in \{0, 1\}$ ensures the correct phase. From this form it is clear that by amplifying the noise on the ancilla systems $|\psi_a\rangle_n$, ZNE can be performed on the main system. More details on the error-mitigation and the data acquisition can be found in appendices F and G.

Note, that comparing the ideal expectation value with the mitigated ones, we achieve an absolute accuracy of roughly 0.01 Ha. While we accomplished promising improvements over the unmitigated results, our mitigated results are still not within the range of 'chemical accuracy' of 0.001 Ha [69]. To be able to make the full calculation with all 14 groups to such accuracy, more sophisticated error-mitigation [70, 71] will be necessary together with a larger shot budget to get better statistics. However, as a proof-of-principle, we believe that this experiment shows that our techniques are very promising for future research.

6. Conclusion and outlook

In this work, we introduced an algorithm that allows the mapping of a sequence of unitaries generated by Pauli strings in the quantum circuit model to a measurement pattern by introducing one ancilla qubit per unitary. We showed that in the case of commuting operators these patterns can be parallelized leading to a constant-depth quantum operation. This result is useful in the NISQ-era, in which quantum circuits have to be as shallow as possible, as well as when fault-tolerant quantum computers are available and one needs to be able to perform unitary operations fast in parallel. The pattern always consists of three layers: a Clifford part, the measurement of the ancilla qubits, and a correction layer that consists of local Pauli operations. Furthermore, we showed that by compressing the Clifford part to an LC-equivalent graph state and simulating the main qubits classically, we can significantly reduce the complexity of a given quantum circuit. We further optimized our graph states using simulated annealing to minimize the hardware requirements for state preparation.

We first applied the algorithm to QAOA. By generating random instances with varying sizes and complexity, we found that there does not seem to be an advantage of using the graph-based ansatz vs. the circuit model for QAOA, since the reduction in depth seems to be limited to smaller problem sizes—although this could be a result of not fully optimized graphs. We found that the simulated annealing algorithm does not reduce the number of edges significantly for difficult instances. It would be interesting to try more sophisticated optimization algorithms in these cases. Furthermore, using embedding techniques to reduce the degree and thus the preparation depth of the graph states [47] could still be beneficial for future applications.

We also ran one specific QAOA example on IBM-hardware. Despite obtaining correct results in the simulation, our experiments on quantum hardware achieved a rather poor fidelity of $F = 0.28$, probably due to errors caused by mid-circuit measurements. To overcome this hurdle on current NISQ-hardware, one could try to design ansätze, that do not require mid-circuit measurements, as we have done for the VQE. One possible direction could be to restrict the mixer-operator space to Clifford operators, i.e. $\{I, H, S\}$, thus removing the necessity for mid-circuit measurements. As has been shown in [72], restricting the QAOA ansatz to the pure Clifford manifold already provides good approximate solutions to the max-cut problem. Combining this ansatz with a measurement pattern consisting of commuting operators only, might be a possible road to find more efficient ansätze.

As a second example, we applied our mapping technique to the VQE using the qubit-ADAPT ansatz [73]. We generated random problem instances of various sizes and complexity and compared the depth of the ansatz in the circuit model with the depth of the graph-based ansatz. For all instances, we found that the graph-based approach improved the depth by a factor of 2 to 5.

We then applied the graph-state conversion algorithm to a specific example using IBM-hardware: the computation of the ground-state energy of H_2O. Here, we specifically designed an ansatz of mutually-commuting operators to avoid mid-circuit measurements. This approach significantly outperforms the standard circuit model approach in terms of circuit depth. We further partitioned the Hamiltonian into mutually-commuting groups of operators, which we were able to measure simultaneously without significantly increasing the overall circuit depth, allowing us to reduce the sampling overhead without any trade-off unlike in the common circuit model. We showed that running our circuits on current IBM hardware, we were able to extract expectation values with an absolute accuracy of roughly 0.01 Ha. This was only possible using error-mitigation techniques to boost the precision of our results. We showed, how such error-mitigation techniques can be incorporated in our formalism by performing ZNE using local gate folding.

We believe that our mapping techniques will provide a useful alternative compared to the standard circuit model for future NISQ- and fault-tolerant algorithms, since it can reduce the overall execution time by employing parallel quantum operations. It is straightforward to apply our techniques to other cases, such as quantum circuits following from trotterized time evolution or other VQE ansätze in different scenarios.

There are several directions for future research. While in this work, we have only shown a finalized qubit-ADAPT-VQE ansatz [29], it is straightforward to apply this algorithm to the entire ADAPT-VQE protocol [74], that is by gradually extending the graph state with each operator added to the ansatz. Our approach for measuring mutually-commuting groups of operators could be efficiently paired with a recent approach on measuring operator gradients in ADAPT-VQE [75]. Next, we want to study how graph states can be implemented more efficiently on a quantum computer by optimizing the graph with respect to its topology thus reducing the preparation depth. By using local complementation and Pauli measurements, hardware-efficient graphs could be implemented which are equivalent to the target state [47]. Additionally, since the number of qubits on a quantum computer is limited, an important line of research is how ancilla qubits can be reused efficiently after their measurement [76]. An interesting alternative approach could be to use circuit-cutting techniques [73] to cut the graph state into several partitions. Here, it is important to cut the circuit corresponding to the measurement pattern at a location, such that the additional sampling overhead is minimal. Last but not least, we want to study how probabilistic error cancellation [70, 71] can be incorporated in our framework.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgment

We thank Kathrin König for sharing her code for zero-noise extrapolation, which we used to perform the local gate-folding. We would also like to thank Jonas Jäger, Daniel Barragan-Yani, Paul Haubenwallner, Timon Scheiber, Johannes S. Mueller-Roemer and André Stork, as well as the anonymous reviewers, for helpful comments on the manuscript. We further acknowledge helpful discussions with Robert Raussendorf, Miriam Backens, Thomas Plehn and Sophia Economou. This work was supported in part by the research project *Zentrum für Angewandtes Quantencomputing* (ZAQC), which is funded by the Hessian Ministry for Digital Strategy and Innovation and the Hessian Ministry of Higher Education, Research and the Arts (TNK and MH), the DLR Quantum Computing Initiative and the Federal Ministry for Economic Affairs and Climate Action; qci.dlr.de/projects/quanticom (TNK). We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

Appendix A. Conditions for parallel measurements of the ancilla qubits

In this appendix we derive equation (9). Consider two Pauli strings $\mathcal{P}$ and $\tilde{\mathcal{P}}$. We want to derive under which conditions we can apply the measurement pattern implementing the unitary

$$U = e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}} e^{-\frac{i}{2}\theta\mathcal{P}} \tag{29}$$

with parallel, i.e. non-adaptive, measurements. We know, that the implementation of the first pattern, which applies the unitary $e^{-\frac{i}{2}\theta\mathcal{P}}$, leads to a Pauli correction given by $\mathcal{P}^s$, where s is the measurement outcome of the ancilla. This correction needs to be shifted through the pattern implementing the second unitary $e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}}$. This can be achieved as follows.

First, we decompose the matrix exponential

$$e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}} = \left[\cos\left(\frac{\tilde{\theta}}{2}\right) - i\sin\left(\frac{\tilde{\theta}}{2}\right)\tilde{\mathcal{P}}\right]. \tag{30}$$

We then shift $\mathcal{P}^s$ through the second unitary by inserting an identity,

$$e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}}\mathcal{P}^s = \mathcal{P}^s\left[\cos\left(\frac{\tilde{\theta}}{2}\right) - i\sin\left(\frac{\tilde{\theta}}{2}\right)\mathcal{P}^s\tilde{\mathcal{P}}\mathcal{P}^s\right]. \tag{31}$$

In case that $[\mathcal{P},\tilde{\mathcal{P}}] = 0$, we have $\mathcal{P}^s\tilde{\mathcal{P}}\mathcal{P}^s = \tilde{\mathcal{P}}$, and thus obtain the trivial commutation relation

$$e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}}\mathcal{P}^s = \mathcal{P}^s e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}}. \tag{32}$$

However, if $[\mathcal{P},\tilde{\mathcal{P}}] \neq 0$, we have $\mathcal{P}^s\tilde{\mathcal{P}}\mathcal{P}^s = (-1)^s\tilde{\mathcal{P}}$. Then, we can rearrange equation (31) as follows:

$$\begin{aligned} e^{-\frac{i}{2}\tilde{\theta}\tilde{\mathcal{P}}}\mathcal{P}^s &= \mathcal{P}^s\left[\cos\left(\frac{\tilde{\theta}}{2}\right) - i\sin\left(\frac{\tilde{\theta}}{2}\right)(-1)^s\tilde{\mathcal{P}}\right] \\ &= \mathcal{P}^s\left[\cos\left((-1)^s\frac{\tilde{\theta}}{2}\right) - i\sin\left((-1)^s\frac{\tilde{\theta}}{2}\right)\tilde{\mathcal{P}}\right] \\ &= \mathcal{P}^s e^{-\frac{i}{2}(-1)^s\tilde{\theta}\tilde{\mathcal{P}}}. \end{aligned} \tag{33}$$

Consequently, by applying our protocol to non-commuting strings, adaptive rotations are introduced, leading to adaptive measurements.

Appendix B. Simulated annealing on graph states

In this appendix we introduce the simulated annealing algorithm we used to simplify graph states. The goal is to minimize a cost function $f: D \to \mathbb{R}$ on a solution space D. The solution space is the set of equivalent graph states in our case.

1. *Initialization*
   Select an initial solution $g \in D$ and a monotonously falling sequence of (positive) temperatures T_i.
2. *Local change*
   For all vertices α of g, perform a local complementation $\tilde{g} \equiv \mathrm{LC}_\alpha(g)$.
   If $f(\tilde{g}) \leqslant f(g)$, set $g = \tilde{g}$. Else, set $g = \tilde{g}$ with a probability of

$$P(g,\tilde{g}) = \exp\left(-\frac{f(\tilde{g}) - f(g)}{T_i}\right). \tag{34}$$

3. *Update*
   If $f(g)$ is better than the previous best solution, update it.
4. *Increment*
   Set $i = i + 1$.
5. *Repeat*
   Repeat steps 2–4 until the final temperature is reached.

Appendix C. Grouping of the H_2O Hamiltonian

In this appendix we show the first four groups of the Hamiltonian of H_2O. As mentioned in section 5.2.3, we used the built-in `qiskit` function `group_commuting()` to find the partitioning. In table 5, we show the terms contributing to H_1 and H_2 and in table 6 the terms contributing to H_3 and H_4.

Finding commuting groups of an Hamiltonian scales exponentially with the system size in general. However, as explained by by Gokhale *et al* [20], for the electronic-structure problem a partitioning of the Hamiltonian, which scales polynomially with the system size, can be performed.

In order to measure the terms from these four groups simultaneously, we first identify an operator basis for each group, from which all other operators can be constructed by multiplication. These bases can be found by mapping the multiplication of Pauli matrices to the group $\mathbf{Z}_2 \otimes \mathbf{Z}_2$. This is achieved by the mapping

$$I \to (0,0), \quad X \to (0,1), \quad Y \to (1,0), \quad Z \to (1,1). \tag{35}$$

Multiplication of two Pauli matrices is thus mapped to addition in $\mathbf{Z}_2 \otimes \mathbf{Z}_2$ (up to a global phase), e.g.

$$XY = (0,1) + (1,1) = (1,0) = Z. \tag{36}$$

Let us now consider a set of m commuting Pauli strings acting on n qubits. From this set, we first construct a $m \times (2n)$ matrix M using the above mapping, i.e. we assign a 1 to the entry $(i, 2j)$, if the ith Pauli string has a Y or Z on the jth position, and a 1 to the entry $(i, 2j+1)$ if it has a X or Y on the jth position. Calculating the left nullspace of M then gives information on which Pauli strings in the set are independent and how they are related by multiplication.

As an example, let us consider the set $\{X_1X_2, Z_1Z_2, Y_1Y_2\}$. The matrix M is given by

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \tag{37}$$

If we calculate the left nullspace of M, we find all possibilities how to add rows of M, such that their sum is zero. In the example, the left nullspace is given by $(1,1,1)$, which means that adding all rows of M yields 0. From this we can infer that

$$(X_1X_2) \cdot (Z_1Z_2) \cdot (Y_1Y_2) \propto 1, \tag{38}$$

which shows that the three Pauli string are not independent, and the expectation value of one of them can be

Table 5. Terms contributing to the Hamiltonians H_1 and H_2 (first two groups). All terms in one table commute with each other.

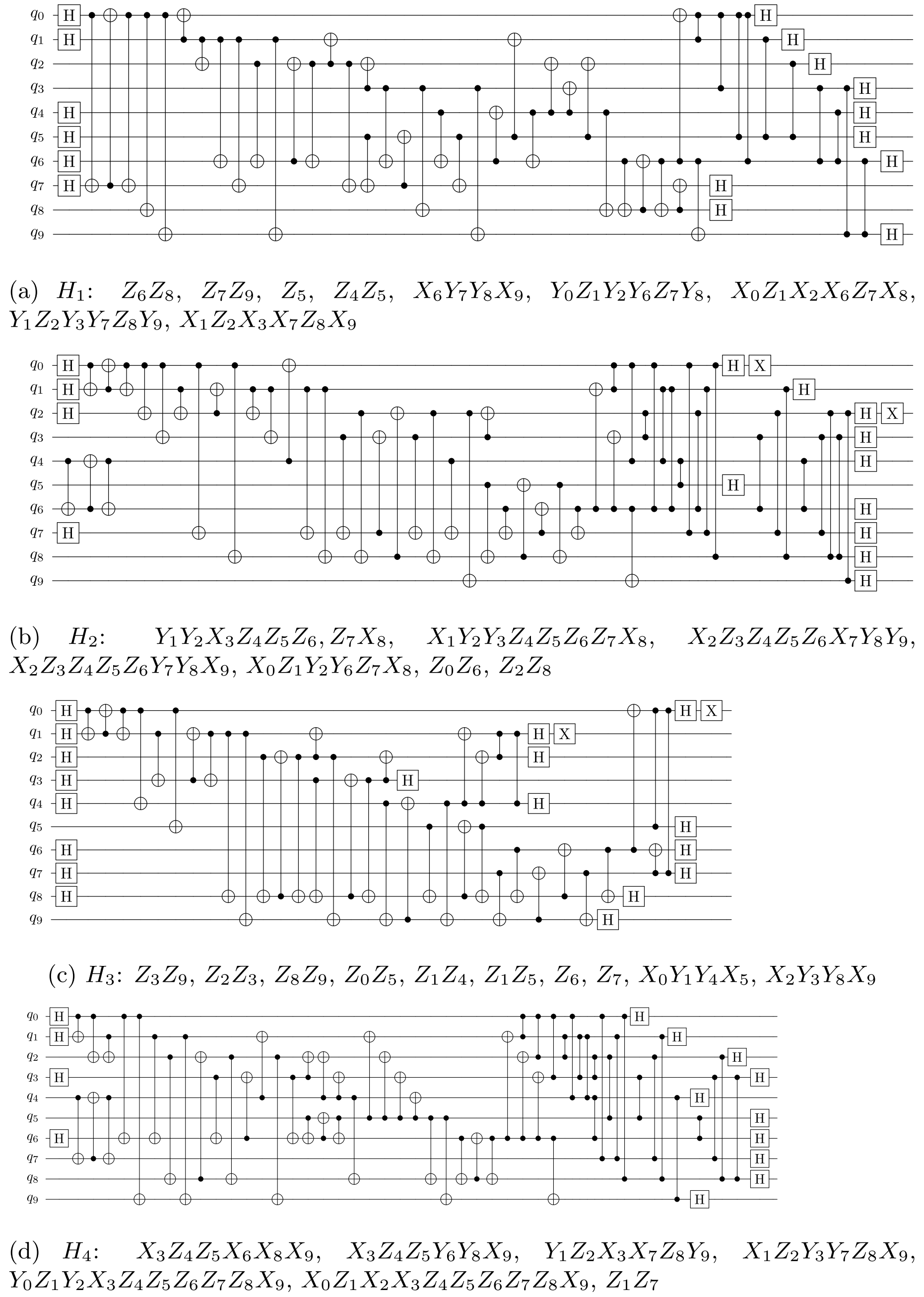
H_1	
Pauli	Coeff.
$Y_0X_1X_2Y_3$	0.01176
$Y_0Y_1X_2X_3$	-0.01176
$X_0X_1Y_2Y_3$	-0.01176
$X_0Y_1Y_2X_3$	0.01176
$Y_0Z_1Y_2Y_7Z_8Y_9$	0.01186
$Y_0Z_1Y_2X_7Z_8X_9$	0.01186
$X_0Z_1X_2Y_7Z_8Y_9$	0.01186
$X_0Z_1X_2X_7Z_8X_9$	0.01186
$Y_1Z_2Y_3Y_6Z_7Y_8$	0.01186
$Y_1Z_2Y_3X_6Z_7X_8$	0.01186
$X_1Z_2X_3Y_6Z_7Y_8$	0.01186
$X_1Z_2X_3X_6Z_7X_8$	0.01186
$Y_6X_7X_8Y_9$	0.02894
$Y_6Y_7X_8X_9$	-0.02894
$X_6X_7Y_8Y_9$	-0.02894
$X_6Y_7Y_8X_9$	0.02894
$Y_0Z_1Y_2X_6Z_7X_8$	0.00076
$X_0Z_1X_2Y_6Z_7Y_8$	0.00076
$Y_1Z_2Y_3X_7Z_8X_9$	0.00076
$X_1Z_2X_3Y_7Z_8Y_9$	0.00076
Z_0Z_2	0.13797
$Y_0Z_1Y_2Y_6Z_7Y_8$	-0.00757
$X_0Z_1X_2X_6Z_7X_8$	-0.00757
Z_1Z_3	0.13797
$Y_1Z_2Y_3Y_7Z_8Y_9$	-0.00757
$X_1Z_2X_3X_7Z_8X_9$	-0.00757
Z_6Z_8	0.1126
Z_7Z_9	0.1126
Z_4	0.48237
Z_5	0.48237
Z_4Z_5	0.22004

H_2	
Pauli	Coeff.
$Y_0X_1X_3Z_4Z_5Y_6$	0.00715
$Y_0Y_1Y_3Z_4Z_5Y_6$	0.00715
$X_0X_1X_3Z_4Z_5X_6$	0.00715
$X_0Y_1Y_3Z_4Z_5X_6$	0.00715
$Y_0Z_1Z_2Z_3Z_4Z_5Y_6Y_7Z_8Y_9$	-0.01627
$Y_0Z_1Z_2Z_3Z_4Z_5Y_6X_7Z_8X_9$	-0.01627
$X_0Z_1Z_2Z_3Z_4Z_5X_6Y_7Z_8Y_9$	-0.01627
$X_0Z_1Z_2Z_3Z_4Z_5X_6X_7Z_8X_9$	-0.01627
$Y_1X_2X_3Z_4Z_5Z_6Z_7Y_8$	0.00064
$Y_1Y_2X_3Z_4Z_5Z_6Z_7X_8$	-0.00064
$X_1X_2Y_3Z_4Z_5Z_6Z_7Y_8$	-0.00064
$X_1Y_2Y_3Z_4Z_5Z_6Z_7X_8$	0.00064
$Y_2Z_3Z_4Z_5Z_6X_7X_8Y_9$	-0.01444
$Y_2Z_3Z_4Z_5Z_6Y_7X_8X_9$	0.01444
$X_2Z_3Z_4Z_5Z_6X_7Y_8Y_9$	0.01444
$X_2Z_3Z_4Z_5Z_6Y_7Y_8X_9$	-0.01444
$Y_0Z_1X_2X_6Z_7Y_8$	-0.00832
$X_0Z_1Y_2Y_6Z_7X_8$	-0.00832
Z_0Z_6	0.12496
Z_2Z_8	0.13512

Table 6. Terms contributing to the Hamiltonians H_3 and H_4.

H_3	
Pauli	Coeff.
$Y_0X_1X_4Y_5$	0.0072
$Y_0Y_1X_4X_5$	-0.0072
$X_0X_1Y_4Y_5$	-0.0072
$X_0Y_1Y_4X_5$	0.0072
$Y_2X_3X_8Y_9$	0.01716
$Y_2Y_3X_8X_9$	-0.01716
$X_2X_3Y_8Y_9$	-0.01716
$X_2Y_3Y_8X_9$	0.01716
Z_2Z_9	0.15228
Z_3Z_8	0.15228
Z_3Z_9	0.13512
Z_0Z_1	0.1583
Z_2Z_3	0.19617
Z_6Z_7	0.14912
Z_8Z_9	0.15503
Z_0Z_4	0.15003
Z_0Z_5	0.15723
Z_1Z_4	0.15723
Z_1Z_5	0.15003
Z_6	0.10364
Z_7	0.10364

H_4	
Pauli	Coeff.
$Y_0X_1X_2Z_3Z_4Z_5Z_6Y_7$	-0.00715
$Y_0Y_1X_2Z_3Z_4Z_5Z_6X_7$	0.00715
$X_0X_1Y_2Z_3Z_4Z_5Z_6Y_7$	0.00715
$X_0Y_1Y_2Z_3Z_4Z_5Z_6X_7$	-0.00715
$Y_0Z_1Y_2Y_3Z_4Z_5Z_6Z_7Z_8Y_9$	0.00064
$Y_0Z_1Y_2X_3Z_4Z_5Z_6Z_7Z_8X_9$	0.00064
$X_0Z_1X_2Y_3Z_4Z_5Z_6Z_7Z_8Y_9$	0.00064
$X_0Z_1X_2X_3Z_4Z_5Z_6Z_7Z_8X_9$	0.00064
$Y_1Z_2Z_3Z_4Z_5X_6X_7Y_8$	-0.01627
$Y_1Z_2Z_3Z_4Z_5Y_6X_7X_8$	0.01627
$X_1Z_2Z_3Z_4Z_5X_6Y_7Y_8$	0.01627
$X_1Z_2Z_3Z_4Z_5Y_6Y_7X_8$	-0.01627
$Y_3Z_4Z_5X_6X_8Y_9$	0.01444
$Y_3Z_4Z_5Y_6Y_8Y_9$	0.01444
$X_3Z_4Z_5X_6X_8X_9$	0.01444
$X_3Z_4Z_5Y_6Y_8X_9$	0.01444
$Y_1Z_2X_3X_7Z_8Y_9$	-0.00832
$X_1Z_2Y_3Y_7Z_8X_9$	-0.00832
Z_1Z_7	0.12496



(a) H_1: Z_6Z_8, Z_7Z_9, Z_5, Z_4Z_5, $X_6Y_7Y_8X_9$, $Y_0Z_1Y_2Y_6Z_7Y_8$, $X_0Z_1X_2X_6Z_7X_8$, $Y_1Z_2Y_3Y_7Z_8Y_9$, $X_1Z_2X_3X_7Z_8X_9$

(b) H_2: $Y_1Y_2X_3Z_4Z_5Z_6, Z_7X_8$, $X_1Y_2Y_3Z_4Z_5Z_6Z_7X_8$, $X_2Z_3Z_4Z_5Z_6X_7Y_8Y_9$, $X_2Z_3Z_4Z_5Z_6Y_7Y_8X_9$, $X_0Z_1Y_2Y_6Z_7X_8$, Z_0Z_6, Z_2Z_8

(c) H_3: Z_3Z_9, Z_2Z_3, Z_8Z_9, Z_0Z_5, Z_1Z_4, Z_1Z_5, Z_6, Z_7, $X_0Y_1Y_4X_5$, $X_2Y_3Y_8X_9$

(d) H_4: $X_3Z_4Z_5X_6X_8X_9$, $X_3Z_4Z_5Y_6Y_8X_9$, $Y_1Z_2X_3X_7Z_8Y_9$, $X_1Z_2Y_3Y_7Z_8X_9$, $Y_0Z_1Y_2X_3Z_4Z_5Z_6Z_7Z_8X_9$, $X_0Z_1X_2X_3Z_4Z_5Z_6Z_7Z_8X_9$, Z_1Z_7

Figure 23. Clifford circuits mapping the basis operators spanning the terms in H_1 (a), H_2 (b), H_3 (c) and H_4 (d) to single-qubit Pauli-Z operators.

inferred by measuring the other two. For the four groups, we chose the following bases:

- H_1: Z_6Z_8, Z_7Z_9, Z_5, Z_4Z_5, $X_6Y_7Y_8X_9$, $Y_0Z_1Y_2Y_6Z_7Y_8$, $X_0Z_1X_2X_6Z_7X_8$, $Y_1Z_2Y_3Y_7Z_8Y_9$, $X_1Z_2X_3X_7Z_8X_9$,
- H_2: $Y_1Y_2X_3Z_4Z_5Z_6, Z_7X_8$, $X_1Y_2Y_3Z_4Z_5Z_6Z_7X_8$, $X_2Z_3Z_4Z_5Z_6X_7Y_8Y_9$, $X_2Z_3Z_4Z_5Z_6Y_7Y_8X_9$, $X_0Z_1Y_2Y_6Z_7X_8$, Z_0Z_6, Z_2Z_8,
- H_3: Z_3Z_9, Z_2Z_3, Z_8Z_9, Z_0Z_5, Z_1Z_4, Z_1Z_5, Z_6, Z_7, $X_0Y_1Y_4X_5$, $X_2Y_3Y_8X_9$,
- H_4: $X_3Z_4Z_5X_6X_8X_9$, $X_3Z_4Z_5Y_6Y_8X_9$, $Y_1Z_2X_3X_7Z_8Y_9$, $X_1Z_2Y_3Y_7Z_8X_9$, $Y_0Z_1Y_2X_3Z_4Z_5Z_6Z_7Z_8X_9$, $X_0Z_1X_2X_3Z_4Z_5Z_6Z_7Z_8X_9$, Z_1Z_7.

Any other operator from the groups in tables 5 and 6 can be written as a product of some operators in the lists.

In the next step, we employ the algorithm by Gokhale *et al* [20] to construct the Clifford circuits in figure 23, which allow the simultaneous measurement of the basis operators following Gokhale *et al* [20]. Effectively, each basis operator from the basis is mapped to a Pauli Z on a specific qubit. Accordingly, the expectation value of products of these basis operators can then be obtained by measuring the expectation value of a Pauli string containing more than one Pauli Z.

Appendix D. Final corrections for the VQE ansatz

In this appendix we show the final Pauli corrections that appear for the VQE ansatz. These follow from equation (10), which gives the Pauli correction $\mathcal{P}$ before the projection to the measurement bases for measuring mutually-commuting operators simultaneously. To derive the final corrections, let C_i denote the Clifford operator, that performs that projection for the different groups, i.e. $i = 1, 2, 3, 4$. Then, the final corrections can be obtained by

$$\mathcal{P}_i = U_i \mathcal{P} U_i^{-1}. \tag{39}$$

We denote by a_i the measurement outcome of the ith ancilla qubit. We find:

$$\begin{aligned}
\mathcal{P}_1 =& (Z_0X_3X_4X_5Y_6X_9)^{a_0}\,(X_1Y_2Z_3X_5X_6X_9)^{a_1}\,(Z_0X_2X_4X_5Y_6X_7X_9)^{a_2}\\
&\times (X_1X_2Y_5X_6)^{a_3}\,(Z_0Z_2Y_3X_4Y_5Y_6)^{a_4}\,(X_1X_3Y_5X_6X_7)^{a_5}\\
&\times (Y_0Z_2Y_3Y_4Y_5Y_6)^{a_6}\,(X_0X_1X_2Z_4Y_5X_6)^{a_7}\,(X_0X_1Y_2Z_3Z_4X_5X_6X_9)^{a_8},\\
\mathcal{P}_2 =& (X_1Y_2X_5X_6X_7X_8X_9)^{a_0}\,(Y_1Z_2Y_3Z_6X_7X_8Y_9)^{a_1}\,(X_0X_1Y_2X_4X_6Y_7Y_8X_9)^{a_2}\\
&\times (Y_0Z_1X_2Z_3Z_4Z_6X_7X_8Z_9)^{a_3}\,(Y_0X_3Z_4X_5X_6X_7X_8)^{a_4}\\
&\times (Z_0Z_1X_2Z_3Y_4X_5Z_6Y_7Y_8Z_9)^{a_5}\,(Y_0Z_1Y_3Z_4X_5Y_6X_7X_8)^{a_6}\\
&\times (Y_0X_2Z_4X_7X_8Z_9)^{a_7}\,(X_1Z_2X_3X_7X_8Y_9)^{a_8},\\
\mathcal{P}_3 =& (X_0Z_1Y_2X_5X_6X_7)^{a_0}\,(Y_1Z_2)^{a_1}\,(Y_0Z_1Y_2X_5Y_6X_7)^{a_2}\\
&\times (X_0X_5X_6X_7Y_8X_9)^{a_3}\,(X_1X_2Y_8X_9)^{a_4}\,(Y_0X_5Y_6X_7Y_8X_9)^{a_5}\\
&\times (X_0Z_1Y_2X_3Y_4X_6X_7Y_8)^{a_6}\,(Y_1Z_2X_3Y_4X_5Y_8)^{a_7}\,(X_0X_3Y_4X_6X_7X_9)^{a_8},\\
\mathcal{P}_4 =& (X_0Y_2X_4X_5X_6X_7X_8)^{a_0}\,(Y_1X_2Z_3X_4X_5Z_6Z_9)^{a_1}\,(X_0X_1Y_2X_5Y_7Y_8)^{a_2}\\
&\times (X_1Y_2Z_3Y_4X_5Z_6Y_9)^{a_3}\,(X_0Z_1X_2Y_4X_5X_6X_7X_8X_9)^{a_4}\\
&\times (Y_2Z_3Z_4X_5Y_6Z_7Z_8Y_9)^{a_5}\,(X_0Z_1X_2X_4X_5Y_6X_7X_8)^{a_6}\\
&\times (X_1Y_2Z_3X_4X_5Z_9)^{a_7}\,(Y_1X_2Z_3Y_4X_5Y_9)^{a_8}.
\end{aligned} \tag{40}$$

Appendix E. Local unitaries for the VQE ansatz

In this appendix we show all local unitaries which, depending on the pre-simulated measurement outcome of the ten main qubits, have to be applied to the graph states shown in figure 22. These local unitaries for the four different graph states are shown in tables 7–10.

The measurement outcomes of the main qubits are encoded as integers. To decode the measured state, the binary form of the number in little-endian convention has to be used, i.e. the number $5 = (000000101)_2$ corresponds to the measurement outcome, for which the first and third qubit were measured in the state $|1\rangle$ and all others in $|0\rangle$.

The unitaries in all tables are given as a list for all nine ancilla qubits in the graph states in the convention of matrix application, i.e. if the local unitary on the first position in the list reads XH, firstly an H gate and then a X gate on the first ancilla qubit has to be applied.

As can be seen in the tables, different simulated counts of the main qubits can lead to the same stabilizer state on the ancilla qubits. In all cases, we find that only 32 different stabilizer states contribute.

Table 7. Local unitaries that have to be applied to the first graph state measuring H_1.

Counts on main qubits	Local unitaries $\{C_0 \ldots C_8\}$
$\{128, 640\}$	$\{XH, S^\dagger YH, YH, I, S^\dagger, Z, S^\dagger YH, YH, S\}$
$\{224, 736\}$	$\{ZH, S^\dagger H, ZH, Z, S^\dagger, Z, S^\dagger H, ZH, S^\dagger\}$
$\{152, 664\}$	$\{ZH, S^\dagger H, ZH, Z, S, Z, S^\dagger H, ZH, S^\dagger\}$
$\{248, 760\}$	$\{XH, S^\dagger YH, YH, I, S, Z, S^\dagger YH, YH, S\}$
$\{20, 532\}$	$\{ZH, S^\dagger H, ZH, I, S, I, S^\dagger H, ZH, S^\dagger\}$
$\{116, 628\}$	$\{XH, S^\dagger YH, YH, Z, S, I, S^\dagger YH, YH, S\}$
$\{12, 524\}$	$\{XH, S^\dagger YH, YH, Z, S^\dagger, I, S^\dagger YH, YH, S\}$
$\{108, 620\}$	$\{ZH, S^\dagger H, ZH, I, S^\dagger, I, S^\dagger H, ZH, S^\dagger\}$
$\{18, 530\}$	$\{XH, S^\dagger H, ZH, X, S^\dagger, I, S^\dagger H, YH, S^\dagger\}$
$\{114, 626\}$	$\{ZH, S^\dagger YH, YH, Y, S^\dagger, I, S^\dagger YH, ZH, S\}$
$\{10, 522\}$	$\{ZH, S^\dagger YH, YH, Y, S, I, S^\dagger YH, ZH, S\}$
$\{106, 618\}$	$\{XH, S^\dagger H, ZH, X, S, I, S^\dagger H, YH, S^\dagger\}$
$\{134, 646\}$	$\{ZH, S^\dagger YH, YH, X, S, Z, S^\dagger YH, ZH, S\}$
$\{230, 742\}$	$\{XH, S^\dagger H, ZH, Y, S, Z, S^\dagger H, YH, S^\dagger\}$
$\{158, 670\}$	$\{XH, S^\dagger H, ZH, Y, S^\dagger, Z, S^\dagger H, YH, S^\dagger\}$
$\{254, 766\}$	$\{ZH, S^\dagger YH, YH, X, S^\dagger, Z, S^\dagger YH, ZH, S\}$
$\{129, 641\}$	$\{YH, S^\dagger YH, YH, Z, S, Z, S^\dagger YH, YH, S^\dagger\}$
$\{225, 737\}$	$\{H, S^\dagger H, ZH, I, S, Z, S^\dagger H, ZH, S\}$
$\{153, 665\}$	$\{H, S^\dagger H, ZH, I, S^\dagger, Z, S^\dagger H, ZH, S\}$
$\{249, 761\}$	$\{YH, S^\dagger YH, YH, Z, S^\dagger, Z, S^\dagger YH, YH, S^\dagger\}$
$\{21, 533\}$	$\{H, S^\dagger H, ZH, Z, S^\dagger, I, S^\dagger H, ZH, S\}$
$\{117, 629\}$	$\{YH, S^\dagger YH, YH, I, S^\dagger, I, S^\dagger YH, YH, S^\dagger\}$
$\{13, 525\}$	$\{YH, S^\dagger YH, YH, I, S, I, S^\dagger YH, YH, S^\dagger\}$
$\{109, 621\}$	$\{H, S^\dagger H, ZH, Z, S, I, S^\dagger H, ZH, S\}$
$\{19, 531\}$	$\{YH, S^\dagger H, ZH, Y, S, I, S^\dagger H, YH, S\}$
$\{115, 627\}$	$\{H, S^\dagger YH, YH, X, S, I, S^\dagger YH, ZH, S^\dagger\}$
$\{11, 523\}$	$\{H, S^\dagger YH, YH, X, S^\dagger, I, S^\dagger YH, ZH, S^\dagger\}$
$\{107, 619\}$	$\{YH, S^\dagger H, ZH, Y, S^\dagger, I, S^\dagger H, YH, S\}$
$\{135, 647\}$	$\{H, S^\dagger YH, YH, Y, S^\dagger, Z, S^\dagger YH, ZH, S^\dagger\}$
$\{231, 743\}$	$\{YH, S^\dagger H, ZH, X, S^\dagger, Z, S^\dagger H, YH, S\}$
$\{159, 671\}$	$\{YH, S^\dagger H, ZH, X, S, Z, S^\dagger H, YH, S\}$
$\{255, 767\}$	$\{H, S^\dagger YH, YH, Y, S, Z, S^\dagger YH, ZH, S^\dagger\}$

Table 8. Local unitaries that have to be applied to the second graph state measuring H_2.

Counts on main qubits	Local unitaries $\{C_0 \ldots C_8\}$
$\{0, 512, 256, 768, 128, 640, 384, 896, 15, 527, 271, 783, 143, 655, 399, 911\}$	$\{ZH, H, ZH, Z, I, Z, ZH, ZH, H\}$
$\{96, 608, 352, 864, 224, 736, 480, 992, 111, 623, 367, 879, 239, 751, 495, 1007\}$	$\{YH, XH, YH, I, Z, Z, YH, YH, H\}$
$\{16, 528, 272, 784, 144, 656, 400, 912, 31, 543, 287, 799, 159, 671, 415, 927\}$	$\{YH, XH, YH, Z, I, Z, ZH, ZH, XH\}$
$\{112, 624, 368, 880, 240, 752, 496, 1008, 127, 639, 383, 895, 255, 767, 511, 1023\}$	$\{ZH, H, ZH, I, Z, Z, YH, YH, XH\}$
$\{72, 584, 328, 840, 200, 712, 456, 968, 71, 583, 327, 839, 199, 711, 455, 967\}$	$\{YH, XH, XH, Z, I, Z, ZH, ZH, XH\}$
$\{40, 552, 296, 808, 168, 680, 424, 936, 39, 551, 295, 807, 167, 679, 423, 935\}$	$\{ZH, H, H, I, Z, Z, YH, YH, XH\}$
$\{88, 600, 344, 856, 216, 728, 472, 984, 87, 599, 343, 855, 215, 727, 471, 983\}$	$\{ZH, H, H, Z, I, Z, ZH, ZH, H\}$
$\{56, 568, 312, 824, 184, 696, 440, 952, 55, 567, 311, 823, 183, 695, 439, 951\}$	$\{YH, XH, XH, I, Z, Z, YH, YH, H\}$
$\{68, 580, 324, 836, 196, 708, 452, 964, 75, 587, 331, 843, 203, 715, 459, 971\}$	$\{ZH, H, H, Z, I, Z, YH, YH, XH\}$
$\{36, 548, 292, 804, 164, 676, 420, 932, 43, 555, 299, 811, 171, 683, 427, 939\}$	$\{YH, XH, XH, I, Z, Z, ZH, ZH, XH\}$
$\{84, 596, 340, 852, 212, 724, 468, 980, 91, 603, 347, 859, 219, 731, 475, 987\}$	$\{YH, XH, XH, Z, I, Z, YH, YH, H\}$
$\{52, 564, 308, 820, 180, 692, 436, 948,$	$\{ZH, H, H, I, Z, Z, ZH, ZH, H\}$

(Continued.)

Table 8. (Continued.)

$59, 571, 315, 827, 187, 699, 443, 955\}$	
$\{12, 524, 268, 780, 140, 652, 396, 908, 3, 515, 259, 771, 131, 643, 387, 899\}$	$\{YH, XH, YH, Z, I, Z, YH, YH, H\}$
$\{108, 620, 364, 876, 236, 748, 492, 1004, 99, 611, 355, 867, 227, 739, 483, 995\}$	$\{ZH, H, ZH, I, Z, Z, ZH, ZH, H\}$
$\{28, 540, 284, 796, 156, 668, 412, 924, 19, 531, 275, 787, 147, 659, 403, 915\}$	$\{ZH, H, ZH, Z, I, Z, YH, YH, XH\}$
$\{124, 636, 380, 892, 252, 764, 508, 1020, 115, 627, 371, 883, 243, 755, 499, 1011\}$	$\{YH, XH, YH, I, Z, Z, ZH, ZH, XH\}$
$\{66, 578, 322, 834, 194, 706, 450, 962, 77, 589, 333, 845, 205, 717, 461, 973\}$	$\{YH, XH, YH, Z, Z, I, ZH, ZH, XH\}$
$\{34, 546, 290, 802, 162, 674, 418, 930, 45, 557, 301, 813, 173, 685, 429, 941\}$	$\{ZH, H, ZH, I, I, I, YH, YH, XH\}$
$\{82, 594, 338, 850, 210, 722, 466, 978, 93, 605, 349, 861, 221, 733, 477, 989\}$	$\{ZH, H, ZH, Z, Z, I, ZH, ZH, H\}$
$\{50, 562, 306, 818, 178, 690, 434, 946, 61, 573, 317, 829, 189, 701, 445, 957\}$	$\{YH, XH, YH, I, I, I, YH, YH, H\}$
$\{10, 522, 266, 778, 138, 650, 394, 906, 5, 517, 261, 773, 133, 645, 389, 901\}$	$\{ZH, H, H, Z, Z, I, ZH, ZH, H\}$
$\{106, 618, 362, 874, 234, 746, 490, 1002, 101, 613, 357, 869, 229, 741, 485, 997\}$	$\{YH, XH, XH, I, I, I, YH, YH, H\}$
$\{26, 538, 282, 794, 154, 666, 410, 922, 21, 533, 277, 789, 149, 661, 405, 917\}$	$\{YH, XH, XH, Z, Z, I, ZH, ZH, XH\}$
$\{122, 634, 378, 890, 250, 762, 506, 1018, 117, 629, 373, 885, 245, 757, 501, 1013\}$	$\{ZH, H, H, I, I, I, YH, YH, XH\}$
$\{6, 518, 262, 774, 134, 646, 390, 902, 9, 521, 265, 777, 137, 649, 393, 905\}$	$\{YH, XH, XH, Z, Z, I, YH, YH, H\}$
$\{102, 614, 358, 870, 230, 742, 486, 998, 105, 617, 361, 873, 233, 745, 489, 1001\}$	$\{ZH, H, H, I, I, I, ZH, ZH, H\}$
$\{22, 534, 278, 790, 150, 662, 406, 918, 25, 537, 281, 793, 153, 665, 409, 921\}$	$\{ZH, H, H, Z, Z, I, YH, YH, XH\}$
$\{118, 630, 374, 886, 246, 758, 502, 1014, 121, 633, 377, 889, 249, 761, 505, 1017\}$	$\{YH, XH, XH, I, I, I, ZH, ZH, XH\}$
$\{78, 590, 334, 846, 206, 718, 462, 974, 65, 577, 321, 833, 193, 705, 449, 961\}$	$\{ZH, H, ZH, Z, Z, I, YH, YH, XH\}$
$\{46, 558, 302, 814, 174, 686, 430, 942, 33, 545, 289, 801, 161, 673, 417, 929\}$	$\{YH, XH, YH, I, I, I, ZH, ZH, XH\}$
$\{94, 606, 350, 862, 222, 734, 478, 990, 81, 593, 337, 849, 209, 721, 465, 977\}$	$\{YH, XH, YH, Z, Z, I, YH, YH, H\}$
$\{62, 574, 318, 830, 190, 702, 446, 958, 49, 561, 305, 817, 177, 689, 433, 945\}$	$\{ZH, H, ZH, I, I, I, ZH, ZH, H\}$

Table 9. Local unitaries that have to be applied to the third graph state measuring H_3.

Counts on main qubits	Local unitaries $\{C_0 \ldots C_8\}$
$\{768\}$	$\{S^\dagger XH, SH, S, S^\dagger XH, Z, S, YH, I, YH\}$
$\{224\}$	$\{S^\dagger H, SH, S, SH, I, S^\dagger, YH, Y, XH\}$
$\{472\}$	$\{S^\dagger H, SH, S, SH, I, S, YH, Y, YH\}$
$\{568\}$	$\{S^\dagger XH, SH, S, S^\dagger XH, Z, S^\dagger, YH, I, XH\}$
$\{4\}$	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, Z, S, ZH, Y, XH\}$
$\{996\}$	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, I, S^\dagger, ZH, I, YH\}$
$\{732\}$	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, I, S, ZH, I, XH\}$
$\{316\}$	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, Z, S^\dagger, ZH, Y, YH\}$
$\{770\}$	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, I, S, ZH, Y, XH\}$
$\{226\}$	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, Z, S^\dagger, ZH, I, YH\}$
$\{474\}$	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, Z, S, ZH, I, XH\}$
$\{570\}$	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, I, S^\dagger, ZH, Y, YH\}$
$\{6\}$	$\{S^\dagger XH, SH, S, S^\dagger XH, I, S, YH, I, YH\}$

(Continued.)

Table 9. (Continued.)

Counts on main qubits	Local unitaries $\{C_0\dots C_8\}$
{998}	$\{S^\dagger H, SH, S, SH, Z, S^\dagger, YH, Y, XH\}$
{734}	$\{S^\dagger H, SH, S, SH, Z, S, YH, Y, YH\}$
{318}	$\{S^\dagger XH, SH, S, S^\dagger XH, I, S^\dagger, YH, I, XH\}$
{769}	$\{S^\dagger H, SH, S, SH, I, S, YH, Y, XH\}$
{225}	$\{S^\dagger XH, SH, S, S^\dagger XH, Z, S^\dagger, YH, I, YH\}$
{473}	$\{S^\dagger XH, SH, S, S^\dagger XH, Z, S, YH, I, XH\}$
{569}	$\{S^\dagger H, SH, S, SH, I, S^\dagger, YH, Y, YH\}$
{5}	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, I, S, ZH, I, YH\}$
{997}	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, Z, S^\dagger, ZH, Y, XH\}$
{733}	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, Z, S, ZH, Y, YH\}$
{317}	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, I, S^\dagger, ZH, I, XH\}$
{771}	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, Z, S, ZH, I, YH\}$
{227}	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, I, S^\dagger, ZH, Y, XH\}$
{475}	$\{S^\dagger H, S^\dagger XH, S^\dagger X, S^\dagger XH, I, S, ZH, Y, YH\}$
{571}	$\{S^\dagger XH, S^\dagger XH, S^\dagger X, SH, Z, S^\dagger, ZH, I, XH\}$
{7}	$\{S^\dagger H, SH, S, SH, Z, S, YH, Y, XH\}$
{999}	$\{S^\dagger XH, SH, S, S^\dagger XH, I, S^\dagger, YH, I, YH\}$
{735}	$\{S^\dagger XH, SH, S, S^\dagger XH, I, S, YH, I, XH\}$
{319}	$\{S^\dagger H, SH, S, SH, Z, S^\dagger, YH, Y, YH\}$

Table 10. Local unitaries that have to be applied to the fourth graph state measuring H_4.

Counts on main qubits	Local unitaries $\{C_0\dots C_8\}$
{0, 512, 256, 768, 128, 640, 384, 896, 15, 527, 271, 783, 143, 655, 399, 911}	$\{YH, XH, HYS, I, Z, S^\dagger HS, H, ZH, XH\}$
{64, 576, 320, 832, 192, 704, 448, 960, 79, 591, 335, 847, 207, 719, 463, 975}	$\{ZH, H, HXS, Z, I, S^\dagger HS, H, ZH, H\}$
{48, 560, 304, 816, 176, 688, 432, 944, 63, 575, 319, 831, 191, 703, 447, 959}	$\{ZH, H, HXS, I, Z, S^\dagger HS, XH, YH, XH\}$
{112, 624, 368, 880, 240, 752, 496, 1008, 127, 639, 383, 895, 255, 767, 511, 1023}	$\{YH, XH, HYS, Z, I, S^\dagger HS, XH, YH, H\}$
{40, 552, 296, 808, 168, 680, 424, 936, 39, 551, 295, 807, 167, 679, 423, 935}	$\{YH, XH, HXS, I, I, S^\dagger HS^\dagger, H, H, XH\}$
{104, 616, 360, 872, 232, 744, 488, 1000, 103, 615, 359, 871, 231, 743, 487, 999}	$\{ZH, H, HYS, Z, Z, S^\dagger HS^\dagger, H, H, H\}$
{24, 536, 280, 792, 152, 664, 408, 920, 23, 535, 279, 791, 151, 663, 407, 919}	$\{ZH, H, HYS, I, I, S^\dagger HS^\dagger, XH, XH, XH\}$
{88, 600, 344, 856, 216, 728, 472, 984, 87, 599, 343, 855, 215, 727, 471, 983}	$\{YH, XH, HXS, Z, Z, S^\dagger HS^\dagger, XH, XH, H\}$
{36, 548, 292, 804, 164, 676, 420, 932, 43, 555, 299, 811, 171, 683, 427, 939}	$\{ZH, H, HYS, I, I, S^\dagger HS^\dagger, H, H, H\}$
{100, 612, 356, 868, 228, 740, 484, 996, 107, 619, 363, 875, 235, 747, 491, 1003}	$\{YH, XH, HXS, Z, Z, S^\dagger HS^\dagger, H, H, XH\}$
{20, 532, 276, 788, 148, 660, 404, 916, 27, 539, 283, 795, 155, 667, 411, 923}	$\{YH, XH, HXS, I, I, S^\dagger HS^\dagger, XH, XH, H\}$
{84, 596, 340, 852, 212, 724, 468, 980, 91, 603, 347, 859, 219, 731, 475, 987}	$\{ZH, H, HYS, Z, Z, S^\dagger HS^\dagger, XH, XH, XH\}$
{12, 524, 268, 780, 140, 652, 396, 908, 3, 515, 259, 771, 131, 643, 387, 899}	$\{ZH, H, HXS, I, Z, S^\dagger HS, H, ZH, H\}$
{76, 588, 332, 844, 204, 716, 460, 972, 67, 579, 323, 835, 195, 707, 451, 963}	$\{YH, XH, HYS, Z, I, S^\dagger HS, H, ZH, XH\}$
{60, 572, 316, 828, 188, 700, 444, 956, 51, 563, 307, 819, 179, 691, 435, 947}	$\{YH, XH, HYS, I, Z, S^\dagger HS, XH, YH, H\}$
{124, 636, 380, 892, 252, 764, 508, 1020, 115, 627, 371, 883, 243, 755, 499, 1011}	$\{ZH, H, HXS, Z, I, S^\dagger HS, XH, YH, XH\}$
{34, 546, 290, 802, 162, 674, 418, 930, 45, 557, 301, 813, 173, 685, 429, 941}	$\{YH, XH, HXS, I, I, S^\dagger HS^\dagger, XH, YH, H\}$
{98, 610, 354, 866, 226, 738, 482, 994,	$\{ZH, H, HYS, Z, Z, S^\dagger HS^\dagger, XH, YH, XH\}$

(Continued.)

Table 10. (Continued.)

109, 621, 365, 877, 237, 749, 493, 1005}	
{18, 530, 274, 786, 146, 658, 402, 914, 29, 541, 285, 797, 157, 669, 413, 925}	$\{ZH, H, HYS, I, I, S^\dagger HS^\dagger, H, ZH, H\}$
{82, 594, 338, 850, 210, 722, 466, 978, 93, 605, 349, 861, 221, 733, 477, 989}	$\{YH, XH, HXS, Z, Z, S^\dagger HS^\dagger, H, ZH, XH\}$
{10, 522, 266, 778, 138, 650, 394, 906, 5, 517, 261, 773, 133, 645, 389, 901}	$\{YH, XH, HYS, I, Z, S^\dagger HS, XH, XH, H\}$
{74, 586, 330, 842, 202, 714, 458, 970, 69, 581, 325, 837, 197, 709, 453, 965}	$\{ZH, H, HXS, Z, I, S^\dagger HS, XH, XH, XH\}$
{58, 570, 314, 826, 186, 698, 442, 954, 53, 565, 309, 821, 181, 693, 437, 949}	$\{ZH, H, HXS, I, Z, S^\dagger HS, H, H, H\}$
{122, 634, 378, 890, 250, 762, 506, 1018, 117, 629, 373, 885, 245, 757, 501, 1013}	$\{YH, XH, HYS, Z, I, S^\dagger HS, H, H, XH\}$
{6, 518, 262, 774, 134, 646, 390, 902, 9, 521, 265, 777, 137, 649, 393, 905}	$\{ZH, H, HXS, I, Z, S^\dagger HS, XH, XH, XH\}$
{70, 582, 326, 838, 198, 710, 454, 966, 73, 585, 329, 841, 201, 713, 457, 969}	$\{YH, XH, HYS, Z, I, S^\dagger HS, XH, XH, H\}$
{54, 566, 310, 822, 182, 694, 438, 950, 57, 569, 313, 825, 185, 697, 441, 953}	$\{YH, XH, HYS, I, Z, S^\dagger HS, H, H, XH\}$
{118, 630, 374, 886, 246, 758, 502, 1014, 121, 633, 377, 889, 249, 761, 505, 1017}	$\{ZH, H, HXS, Z, I, S^\dagger HS, H, H, H\}$
{46, 558, 302, 814, 174, 686, 430, 942, 33, 545, 289, 801, 161, 673, 417, 929}	$\{ZH, H, HYS, I, I, S^\dagger HS^\dagger, XH, YH, XH\}$
{110, 622, 366, 878, 238, 750, 494, 1006, 97, 609, 353, 865, 225, 737, 481, 993}	$\{YH, XH, HXS, Z, Z, S^\dagger HS^\dagger, XH, YH, H\}$
{30, 542, 286, 798, 158, 670, 414, 926, 17, 529, 273, 785, 145, 657, 401, 913}	$\{YH, XH, HXS, I, I, S^\dagger HS^\dagger, H, ZH, XH\}$
{94, 606, 350, 862, 222, 734, 478, 990, 81, 593, 337, 849, 209, 721, 465, 977}	$\{ZH, H, HYS, Z, Z, S^\dagger HS^\dagger, H, ZH, H\}$

Appendix F. Zero-Noise Extrapolation for measurement patterns

ZNE is an error-mitigation strategy, in which a given quantum circuit is artificially stretched to amplify the noise [67]. In order to mitigate expectation values, we first measure the observable at different noise amplification levels λ and performs an extrapolation to the zero-noise limit.

In our measurement-based approach we can use zero-noise mitigation directly on the final observable. Following Dehaene and De Moor [52], the state before measurement of the ancilla and main qubits can be written as:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_n c_n \left(\prod_{i\in\mathcal{C}} \prod_{j\in\mathcal{N}_i} CP_{ji} \right) |\psi_a\rangle_n |\psi_q\rangle_n, \tag{41}$$

where $c_n \in \{\pm 1, \pm i\}$ and N is the number of classical measurement outcomes, $|\psi_q\rangle_n$ denotes the nth possible computational state of the main register, $|\psi_a\rangle_n$ corresponds to the quantum state of the ancilla circuit, $\mathcal{C}$ is the set of main qubits, $\mathcal{N}_i$ is the set of ancilla qubits connected to the ith main qubit, and CP_{ji} denotes the entangling Pauli gates between both states, that can also be performed by classical post-processing the measurement outcomes. Note that the controlled Pauli gates CP_{ji} are thus always controlled by the ancilla qubits.

Without loss of generality, we may assume that only Z-expecation values have to be measured on the main register (any basis change can be absorbed into the definition of the state in equation (41)). Let $\mathcal{Z}_q \in \{I, Z\}^N$, then:

$$\langle\psi|\mathcal{Z}_q|\psi\rangle = \frac{1}{N} \sum_{n,m} c_m^* c_{nm} \langle\psi_a|_m \langle\psi_q| \left(\prod_{i\in\mathcal{C}} \prod_{j\in\mathcal{N}_i} CP_{ji}\ \mathcal{Z}_q\ CP_{ji} \right) |\psi_a\rangle_n |\psi_q\rangle_n. \tag{42}$$

Using the properties of the Clifford group (cf equation (8)), we may drag the Pauli corrections CP_{ji} across the Pauli string $\mathcal{Z}_q$:

$$CX_{ji}\left(I_j\otimes Z_i\right)CX_{ji}=Z_j\otimes Z_i. \tag{43}$$

Consequently, we may rewrite

$$\prod_{i\in\mathcal{C}}\prod_{j\in\mathcal{N}_i}CP_{ji}\mathcal{Z}_qCP_{ji}=\mathcal{Z}_a\otimes\mathcal{Z}_q, \tag{44}$$

where $\mathcal{Z}_a$ is the Pauli string acting on the ancilla register. This step effectively transforms the Pauli string $\mathcal{Z}_c$, which was previously only acting on the main register, to an operator, that acts also on the ancilla register. By inserting equation (44) into equation (42), we obtain

$$\langle\psi|\mathcal{Z}_q|\psi\rangle=\frac{1}{N}\sum_{n,m}c_m^*c_{nm}\langle\psi_a|\mathcal{Z}_a|\psi_a\rangle_{nm}\langle\psi_q|\mathcal{Z}_q|\psi_q\rangle_n. \tag{45}$$

Next, we can exploit that the computational states $|\psi_q\rangle_n$ are eigenstates of the Pauli-Z operator, hence

$${}_m\langle\psi_q|\mathcal{Z}_q|\psi_q\rangle_n=(-1)^{s_n}\,\delta_{nm}, \tag{46}$$

where $s_n\in\{0,1\}$ ensures the correct phase according to the bitstring. This leads us to the final expression

$$\langle\psi|\mathcal{Z}_q|\psi\rangle=\frac{1}{N}\sum_n(-1)^{s_n}\,{}_n\langle\psi_a|\mathcal{Z}_a|\psi_a\rangle_n. \tag{47}$$

Equation (47) shows, that the expectation value of an observable $\mathcal{Z}_q$ on the main qubits can be written as a sum of Z expectation values on the ancilla qubits. The relationship between $\langle\mathcal{Z}_q\rangle$ and the sum over $\langle\mathcal{Z}_a\rangle$ shows, why zero noise extrapolation works in our approach. From equation (47) we see that amplifying the noise in the $|\psi_a\rangle_n$ states will increase the noise in the expectation values given by ${}_n\langle\psi_a|\mathcal{Z}_a|\psi_a\rangle_n$. Since all these states are equivalent up to local unitaries for all n, we can increase the noise of $\langle\mathcal{Z}_q\rangle$ in a well-defined way by amplifying the noise in the circuits which prepare the ancilla states $|\psi_a\rangle_n$.

Appendix G. Data acquisition for the VQE experiment

In this appendix, we show how we performed the ZNE for the case of the VQE from section 5.2.3. In order to increase the noise in the ancilla circuits, we use the method of local gate folding [68]. In this method, all CNOT gates in the circuit are replaced as

$$CX_{ij}\to CX_{ij}^{2n+1} \tag{48}$$

where n is an integer. The additional CNOT gates do not change the outcome of the circuit, but stretch the pulse which is executed on the hardware. This leads to a higher error-rate due to decoherence effects in the qubits, which is the main source of errors. In order to perform ZNE, we estimate λ by calculating the factor by which the pulse is stretched in time, see figure 24.

In addition to ZNE we use Pauli twirling [65] and dynamical decoupling [66] as additional mitigation techniques. For the Pauli twirling, we substitute each CNOT gate in a given circuit randomly by

$$CX_{ij}\to P_{1i}P_{2j}CX_{ij}P_{3i}P_{4j}, \tag{49}$$

where $P_i=X,Z,Y$ or I, and where P_1 and P_2 are chosen randomly and P_3 and P_4 such, that the circuit do not change the effect of the original gate. In total, there are 16 different combinations how to substitute the CNOT gate.

In figure 25 we show the results from running our circuits on `ibm_hanoi`. Before running the graph state circuits, we first ran read-out mitigation circuits using the `m3` package using 100 000 shots per circuit. By calculating the readout-calibration matrix, we are then able to perform readout-error mitigation on the results. As can be seen from figure 25, read-out errors are an important source of error in our algorithm and need to be corrected before performing ZNE.

We evaluate the expectation values at three different noise levels, $\lambda\in\{1,3,5\}$. In order to extrapolate to the zero-noise limit at $\lambda=0$ we use a second-order polynomial fit:

$$\langle H(\lambda)\rangle=a\lambda^2+b\lambda+c, \tag{50}$$

such that in the zero-noise limit the expectation value is given by c and its error by the fitting error. We ran each experiment eight times to quantify the effect of statistical shot noise.

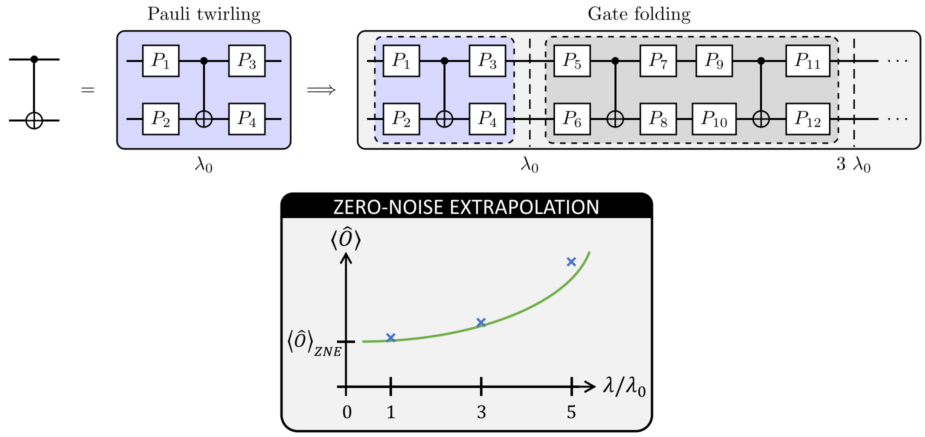


Figure 24. Overview of the error-mitigation strategies we performed. Upper panel: in order to amplify the noise, we can substitute one CNOT gate by three. In addition, we use randomized compiling over the Pauli group (a.k.a Pauli twirling) to mitigate coherent noise [65]. Lower panel: After measuring the observable at different noise levels, we can perform an extrapolation the zero-noise limit.

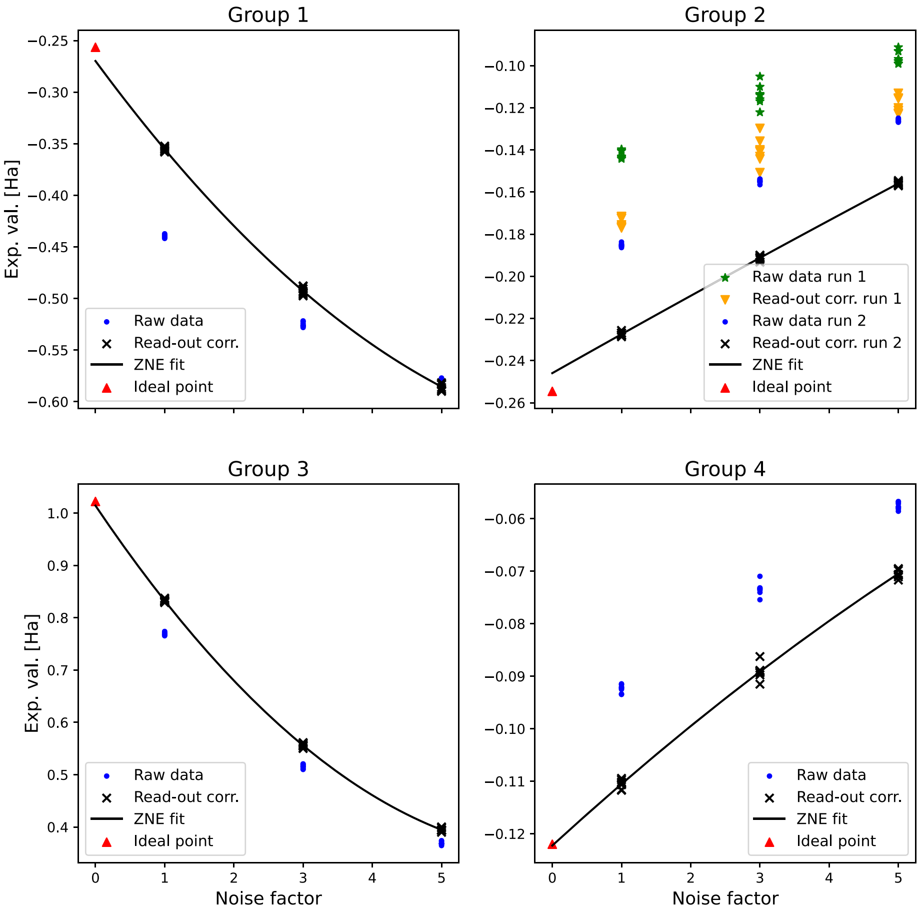


Figure 25. Data acquisition for the ZNE for measuring the expectation values of the first four groups of the partitioning of the H_2O Hamiltonian. We ran each experiment eight times to see the statistical effect of shot noise. We performed read-out error mitigation using the `M3` package. In all cases, we find that read-out error mitigation together with ZNE significantly improve the extracted expectation value comparing with the ideal ones (red triangles in the plot). We performed the measurement of the expectation value of the second group twice, since the first run gave inconsistent results. This can be seen in the data points which are distributed with a much higher standard deviation compared to the data of the other runs. The second run gave much better results, although it ran just roughly 60 minutes later.

ORCID iDs

Thierry N Kaldenbach https://orcid.org/0009-0008-5607-4427
Matthias Heller https://orcid.org/0000-0002-4774-5072

References

[1] Raussendorf R and Briegel H J 2001 A one-way quantum computer *Phys. Rev. Lett.* **86** 5188
[2] Jozsa R 2006 An introduction to measurement based quantum computation *NATO Science Series, III: Computer and Systems Sciences. Quantum Information Processing-From Theory to Experiment* vol 199 pp 137–58
[3] Briegel H J and Raussendorf R 2001 Persistent entanglement in arrays of interacting particles *Phys. Rev. Lett.* **86** 910–3
[4] Broadbent A and Kashefi E 2009 Parallelizing quantum circuits *Theor. Comput. Sci.* **410** 2489–510
[5] Ferguson R R, Dellantonio L, Balushi A A, Jansen K, Dür W and Muschik C A 2021 Measurement-based variational quantum eigensolver *Phys. Rev. Lett.* **126** 220501
[6] Miyazaki J, Hajdušek M and Murao M 2015 Analysis of the trade-off between spatial and temporal resources for measurement-based quantum computation *Phys. Rev.* A **91** 052302
[7] Backens M, Miller-Bakewell H, de Felice G, Lobski L and van de Wetering J 2021 There and back again: a circuit extraction tale *Quantum* **5** 421
[8] Eslamy M, Houshmand M, Zamani M S and Sedighi M 2018 Optimization of one-way quantum computation measurement patterns *Int. J. Theor. Phys.* **57** 3296–317
[9] Danos V and Kashefi E 2006 Determinism in the one-way model *Phys. Rev.* A **74** 052310
[10] Browne D E, Kashefi E, Mhalla M and Perdrix S 2007 Generalized flow and determinism in measurement-based quantum computation *New J. Phys.* **9** 250
[11] Duncan R and Perdrix S 2010 Rewriting measurement-based quantum computations with generalised flow *International Colloquium on Automata, Languages and Programming* (Springer) pp 285–96
[12] Simmons W 2021 Relating measurement patterns to circuits via pauli flow (arXiv:2109.05654)
[13] van de Wetering J 2020 Zx-calculus for the working quantum computer scientist (arXiv:2012.13966)
[14] Staudacher K, Guggemos T, Grundner-Culemann S and Gehrke W 2023 Reducing 2-qubit gate count for zx-calculus based quantum circuit optimization *Electron. Proc. Theor. Comput. Sci.* **394** 29–45
[15] Kissinger A and van de Wetering J 2020 Reducing the number of non-clifford gates in quantum circuits *Phys. Rev.* A **102** 022406
[16] Cowtan A, Dilkes S, Duncan R, Simmons W and Sivarajah S 2020 Phase gadget synthesis for shallow circuits *EPTCS* **318** 213–28
[17] Chan A, Shi Z, Dellantonio L, Dür W and Muschik C A 2023 Hybrid variational quantum eigensolvers: merging computational models (arXiv:2305.19200)
[18] Mhalla M and Perdrix S 2008 Finding optimal flows efficiently *International Colloquium on Automata, Languages and Programming* (Springer) pp 857–68
[19] Crawford O, Straaten B V, Wang D, Parks T, Campbell E and Brierley S 2021 Efficient quantum measurement of Pauli operators in the presence of finite sampling error *Quantum* **5** 385
[20] Gokhale P, Angiuli O, Ding Y, Gui K, Tomesh T, Suchara M, Martonosi M and Chong F T 2020 $o(n^3)$ measurement cost for variational quantum eigensolver on molecular hamiltonians *IEEE Trans. Quantum Eng.* **1** 1–24
[21] Anders S and Briegel H J 2006 Fast simulation of stabilizer circuits using a graph-state representation *Phys. Rev.* A **73** 022334
[22] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 Optimization by simulated annealing *Science* **220** 671–80
[23] Van Laarhoven P J, Aarts E H, van Laarhoven P J and Aarts E H 1987 *Simulated Annealing* (Springer)
[24] Preskill J 2018 Quantum computing in the nisq era and beyond *Quantum* **2** 79
[25] Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love P J, Aspuru-Guzik A and O'brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213
[26] Tilly J *et al* 2022 The variational quantum eigensolver: a review of methods and best practices *Phys. Rep.* **986** 1–128
[27] Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)
[28] Zhou L, Wang S-T, Choi S, Pichler H and Lukin M D 2020 Quantum approximate optimization algorithm: performance, mechanism and implementation on near-term devices *Phys. Rev.*X **10** 021067
[29] Tang H L, Shkolnikov V, Barron G S, Grimsley H R, Mayhall N J, Barnes E and Economou S E 2021 Qubit-adapt-vqe: an adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor *PRX Quantum* **2** 020310
[30] Shor P W 1996 Fault-tolerant quantum computation *Proc. 37th Conf. on Foundations of Computer Science* (IEEE) pp 56–65
[31] Nielsen M A and Chuang I 2002 *Quantum Information and Quantum Computation* (Cambridge University Press)
[32] Gui K, Tomesh T, Gokhale P, Shi Y, Chong F T, Martonosi M and Suchara M 2020 Term grouping and travelling salesperson for digital quantum simulation (arXiv:2001.05983)
[33] Raussendorf R and Briegel H J 2002 Computational model underlying the one-way quantum computer *Quantum Inf. Comput.* **2** 443–86
[34] Raussendorf R, Browne D E and Briegel H J 2003 Measurement-based quantum computation on cluster states *Phys. Rev.* A **68** 022312
[35] Raussendorf R 2009 Measurement-based quantum computation with cluster states *Int. J. Quantum Inf.* **7** 1053–203
[36] Duncan R, Kissinger A, Perdrix S and van de Wetering J 2020 Graph-theoretic simplification of quantum circuits with the zx-calculus *Quantum* **4** 279
[37] Gottesman D and Chuang I L 1999 Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations *Nature* **402** 390–3
[38] Vijayan M K, Paler A, Gavriel J, Myers C R, Rohde P P and Devitt S J 2022 Compilation of algorithm-specific graph states for quantum circuits *Quantum Sci. Technol.* **9** 2
[39] Herr D, Nori F and Devitt S J 2017 Lattice surgery translation for quantum computation *New J. Phys.* **19** 013034
[40] Paler A, Devitt S J, Nemoto K and Polian I 2014 Mapping of topological quantum circuits to physical hardware *Sci. Rep.* **4** 4657
[41] Paler A, Polian I, Nemoto K and Devitt S J 2017 Fault-tolerant, high-level quantum circuits: form, compilation and description *Quantum Sci. Technol.* **2** 025003
[42] Saadatmand S N *et al* 2024 Fault-tolerant resource estimation using graph-state compilation on a modular superconducting architecture (arXiv:2406.06015)

[43] Bravyi S, Smith G and Smolin J A 2016 Trading classical and quantum computational resources *Phys. Rev.* X **6** 021043

[44] Peres F C R and Galvão E F 2023 Quantum circuit compilation and hybrid computation using pauli-based computation *Quantum* **7** 1126

[45] Hein M, Eisert J and Briegel H J 2004 Multiparty entanglement in graph states *Phys. Rev.* A **69** 062311

[46] Hein M, Dür W, Eisert J, Raussendorf R, Nest M and Briegel H-J 2006 Entanglement in graph states and its applications (arXiv:quant-ph/0602096)

[47] Høyer P, Mhalla M and Perdrix S 2006 Resources required for preparing graph states *Algorithms and Computation* ed T Asano (Springer) pp 638–49

[48] Van den Nest M, Dehaene J and De Moor B 2004 Graphical description of the action of local clifford transformations on graph states *Phys. Rev.* A **69** 022316

[49] Schlingemann D 2001 Stabilizer codes can be realized as graph codes (arXiv:quant-ph/0111080)

[50] Gottesman D 1998 The heisenberg representation of quantum computers (arXiv:quant-ph/9807006)

[51] Treinish M, Carvalho I, Tsilimigkounakis G and Sá N 2022 Rustworkx: a high-performance graph library for python *J. Open Source Softw.* **7** 3968

[52] Dehaene J and De Moor B 2003 Clifford group, stabilizer states and linear and quadratic operations over gf (2) *Phys. Rev.* A **68** 042318

[53] Adcock J C, Morley-Short S, Dahlberg A and Silverstone J W 2020 Mapping graph state orbits under local complementation *Quantum* **4** 305

[54] Qiskit Contributors 2023 Quantum computing with Qiskit (https://doi.org/10.48550/arXiv.2405.08810)

[55] Bian Z, Chudak F, Macready W G, Clark L and Gaitan F 2013 Experimental determination of ramsey numbers *Phys. Rev. Lett.* **111** 130505

[56] Pelofske E, Bärtschi A and Eidenbenz S 2023 Quantum annealing vs. qaoa: 127 qubit higher-order ising problems on nisq computers *Int. Conf. on High Performance Computing* (Springer) pp 240–58

[57] Lubinski T, Johri S, Varosy P, Coleman J, Zhao L, Necaise J, Baldwin C H, Mayer K and Proctor T 2023 Application-oriented performance benchmarks for quantum computing *IEEE Transactions on Quantum Engineering* **4**

[58] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 The theory of variational hybrid quantum-classical algorithms *New J. Phys.* **18** 023023

[59] Ryabinkin I G, Yen T-C, Genin S N and Izmaylov A F 2018 Qubit coupled-cluster method: a systematic approach to quantum chemistry on a quantum computer *J. Chem. Theory Comput.* **14** 12

[60] Xia R and Kais S 2020 Qubit coupled cluster singles and doubles variational quantum eigensolver ansatz for electronic structure calculations *Quantum Sci. Technol.* **6** 015001

[61] Nam Y *et al* 2020 Ground-state energy estimation of the water molecule on a trapped-ion quantum computer *npj Quantum Inf.* **6** 33

[62] McClean J R *et al* 2020 Openfermion: the electronic structure package for quantum computers *Quantum Sci. Technol.* **5** 034014

[63] Sun Q *et al* 2018 Pyscf: the python-based simulations of chemistry framework *WIREs Comput. Mol. Sci.* **8** e1340

[64] Nation P D, Kang H, Sundaresan N and Gambetta J M 2021 Scalable mitigation of measurement errors on quantum computers *PRX Quantum* **2** 040326

[65] Wallman J J and Emerson J 2016 Noise tailoring for scalable quantum computation via randomized compiling *Phys. Rev.* A **94** 052325

[66] Viola L and Lloyd S 1998 Dynamical suppression of decoherence in two-state quantum systems *Phys. Rev.* A **58** 2733–44

[67] Temme K, Bravyi S and Gambetta J M 2017 Error mitigation for short-depth quantum circuits *Phys. Rev. Lett.* **119** 180509

[68] Giurgica-Tiron T, Hindy Y, LaRose R, Mari A and Zeng W J 2020 Digital zero noise extrapolation for quantum error mitigation *2020 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)* (IEEE) pp 306–16

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

[70] Van Den Berg E, Minev Z K, Kandala A and Temme K 2023 Probabilistic error cancellation with sparse pauli–lindblad models on noisy quantum processors *Nat. Phys.* **19** 1–6

[71] Gupta R S, van den Berg E, Takita M, Temme K and Kandala A 2023 Probabilistic error cancellation for measurement-based quantum circuits (arXiv:2310.07825)

[72] Muñoz-Arias M H, Kourtis S and Blais A 2023 Low-depth clifford circuits approximately solve maxcut (arXiv:2310.15022)

[73] Tang W, Tomesh T, Suchara M, Larson J and Martonosi M 2021 Cutqc: using small quantum computers for large quantum circuit evaluations *Proc. 26th ACM Int. Conf. on Architectural Support for Programming Languages and Operating Systems* pp 473–86

[74] Grimsley H R, Economou S E, Barnes E and Mayhall N J 2019 An adaptive variational algorithm for exact molecular simulations on a quantum computer *Nat. Commun.* **10** 3007

[75] Anastasiou P G, Mayhall N J, Barnes E and Economou S E 2023 How to really measure operator gradients in adapt-vqe (arXiv:2306.03227)

[76] Ruh J and Devitt S 2024 Quantum circuit optimisation and mbqc scheduling with a pauli tracking library (arXiv:2405.03970)