Hardware-efficient entangled measurements for variational quantum algorithms

Francisco Escudero,[1,2] David Fernández-Fernández,[1,3] Gabriel Jaumà,[1] Guillermo F. Peñas,[1] and Luciano Pereira[1,*]

[1] *Instituto de Física Fundamental, IFF-CSIC, Calle Serrano 113b, 28006 Madrid, Spain*
[2] *CWI & QuSoft, Science Park 123, 1098 XG Amsterdam, The Netherlands*
[3] *Instituto de Ciencia de Materiales de Madrid, ICMM-CSIC, 28049 Madrid, Spain*

Variational algorithms have received significant attention in recent years due to their potential to solve practical problems in noisy intermediate-scale quantum (NISQ) devices. A fundamental step of these algorithms is the evaluation of the expected value of Hamiltonians, and hence, efficient schemes to perform this task are required. The standard approach employs local measurements of Pauli operators and requires a large number of circuits. An alternative is to make use of entangled measurements, which significantly reduces the number of circuits if the measurement involves entangling gates between nonphysically connected qubits, introducing additional intermediate entangling operations that increase the depth of the circuits. As a solution to this problem we propose hardware-efficient entangled measurements (HEEM), that is, measurements that only permit entanglement between physically connected qubits. We show that this strategy enhances the evaluation of molecular Hamiltonians in NISQ devices by reducing the number of circuits required without increasing their depth. We provide quantitative metrics of how this approach offers better results than local measurements and arbitrarily entangled measurements. We estimate the ground state energy of the H_2O molecule with classical simulators and quantum hardware using the variational quantum eigensolver with HEEM.

I. INTRODUCTION.

We are currently in the era of noisy intermediate-scale quantum computers (NISQ). The main limitations of these devices are short coherence times and noisy entanglement gates; therefore, NISQ circuits must inevitably have a low depth [1]. Given that, a lot of effort has been devoted to the design and implementation of quantum algorithms that only use low-depth circuits [2–5]. Among such algorithms, one of the families that has gained attention are the variational quantum algorithms (VQAs) [6–8]: hybrid quantum-classical methods where a classical computer guides a quantum computer to produce variational quantum states and to measure their expected value, with the goal of minimizing an objective function encoded in a Hamiltonian. The most famous VQAs are the variational quantum eigensolver (VQE) [9] and the quantum approximate optimization algorithm (QAOA) [10]. Multiple VQAs have been applied in a wide range of areas such as chemistry [11–14], finance [15, 16], traffic prediction [17], machine learning [18–21], entanglement detection [22–24], and differential equations [25–27].

Despite the great advances made in recent years, implementing VQA remains a challenge. One of the main drawbacks is the large number of measurements required to evaluate the objective function. Such evaluation can be done by decomposing the Hamiltonian on a basis of tensor products of Pauli operators, also called Pauli strings, and then measuring each of these terms independently. Using this approach, the number of measurements needed to evaluate the objective function is equal to the Pauli terms of the Hamiltonian, which scales as N^4 for typical instances such as second-quantized chemical Hamiltonians on N qubits [28]. The preparation and measurement of each circuit require a non-negligible amount of time and resources; thus, decreasing the number of circuits is essential to speed up VQAs and to reach realistic applications of quantum computing.

Several methods have been proposed to reduce the total number of measurements required to efficiently measure observables, such as classical shadows [29, 30], quantum tomography [31–33], or machine learning [34–36], among others [37–40]. Other proposals are grouping methods, which exploit the commutative relation between the Pauli strings to make groups that can be measured simultaneously, reducing the total number of required experiments. The most widely known approach is the grouping with tensor product basis (TPB) [12–14, 41, 42], which use qubit-wise commutativity and require no entanglement. Another alternative is to use entangled measurements, which further reduces the number of measurements compared to TPB [28, 43–45]. However, this last approach assumes unlimited entanglement resources and is not suitable for NISQ devices. There is a midpoint between the two extreme alternatives of assuming limitless entanglement or none at all: using measurements whose entanglement requirements are within the limits of NISQ devices. There are several works in this line [46–49], nevertheless, none of these techniques takes into account the connectivity of the particular quantum processor where the algorithm is run.

In this article, we address the problem of grouping Pauli strings with entangled measurements (EMs), but only between physically connected qubits, that is, hardware-efficient entangled measurements (HEEMs). Given a set of Pauli strings and a processor connectivity, there is a vast number of possible HEEMs, some of which will be more effective than others. Finding the optimal

* luciano.ivan@iff.csic.es

HEEM requires, among other things, finding the optimal mapping between the theoretical qubits of the algorithm and the physical qubits of the device. We have named this issue the *processor embedding* problem. We propose heuristic algorithms to solve the *processor embedding* problem and to perform the grouping with HEEMs. We run VQE with HEEM to estimate the ground state energy of the H_2O molecule using both classical simulators and quantum hardware. The proposed method reduces the number of measurements due to the entangled measurements, and avoids long-range qubit interactions that would involve noisy, deep circuits, thanks to the hardware-efficient approach.

II. HARDWARE-EFFICIENT GROUPING

A general N-qubit Hamiltonian can be expanded in terms of Pauli strings as

$$H = \sum_{\alpha} h_\alpha P_\alpha, \tag{1}$$

where P_α are the tensor products of the identity and Pauli operators (I, X, Y, Z), and $h_\alpha \in \mathbb{R}$ are the coefficients of each Pauli string. The standard routine to compute the expected value of H consists of measuring each Pauli string P_i with a TPB measurement, which are the tensor products of the bases of eigenstates of the Pauli operators $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$, with $\mathcal{X} = \{|0\rangle \pm |1\rangle\}$, $\mathcal{Y} = \{|0\rangle \pm i\,|1\rangle\}$ and $\mathcal{Z} = \{|0\rangle, |1\rangle\}$.

Multiple Pauli strings can be evaluated simultaneously with a single TPB. In this case, we say that they are *compatible* with that TPB. For example, the 3-qubit Pauli strings XIZ and XYZ are compatible with TPB $\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Z}$. This property allows us to define a grouping method [41], which consists of searching for the smallest set of TPB that can be used to evaluate the expected value of H. Finding the best TPB grouping for a given Hamiltonian is equivalent to finding the best coloring for its Pauli graph, which is NP-Complete [42, 50].

One can go beyond the TPB grouping and measure the expected value of H with fewer groups thanks to EM, that is, measuring after an entangling operation. In [46] the authors proposed a heuristic algorithm to construct groups using EM between pairs of qubits (see Appendix A). The EM grouping proved to be more efficient than the TPB grouping, significantly reducing the number of measurements and the uncertainty on the evaluation of the observables. One drawback of this approach is that it neglects the error of entangling gates in NISQ devices, and hence it might actually worsen the results in some scenarios. This is particularly pronounced when performing entangling operations between nonphysically connected qubits, since these require the usage of mediating qubit—and hence additional entangling gates—that increase the depth and thus the error of the circuits.

As a solution to this problem, we propose three heuristic algorithms to group Pauli strings with HEEM (see Appendix B). These algorithms introduce two important improvements to the algorithm proposed by Hamamura *et al.* [46]. First, the algorithms check whether two Pauli strings are simultaneously measurable by an EM given the connectivity of a device (Algorithm 2), isolating from all possible EM only those that are hardware efficient. Second, since the groups obtained by our algorithms or by Hamamura's algorithms depend on the order followed by several loops, we introduce three algorithms with different ordering between each order. This is explained in the next section.

A. The processor embedding problem

There are multiple factors that determine the performance of the grouping algorithm, for instance, the order that its loops follow to run through the qubits, the assignable measurements, and the Pauli strings. These orders can be optimized considering the *processor embedding* problem: the problem of finding the optimal way to map the theoretical qubits of the Hamiltonian into the physical qubits of the device. In this section, we introduce a heuristic approach to solve this problem, which in turn allows us to optimize the order of the loops in the qubits and in the assignable measurements.

Given the Hamiltonian shown in Eq. (1), its *compatibility matrix* C is defined as a $N \times N$ symmetric matrix whose non-diagonal entries C_{ij} are equal to the number of compatible entangled measurements (see Table II in the Supp. Material) involving qubits i and j. The diagonal entries of C are not relevant for the *processor embedding* problem.

The number of compatible entangled measurements that involve qubits (i, j) is defined in the following way. The factors i and j of each Pauli string P_α can be regarded as a set of Pauli sub-strings of 2 qubits, $\{P_\alpha^{i,j}\}$, and the number of compatible entangled measurements that involve qubits (i, j) is the number of pairs of Pauli sub-strings $\{P_\alpha^{ij}\}$ that are compatible through an entangled measurement. Thus, the compatibility matrix encapsulates how many entangled measurements can be established between two qubits. As an example, let's consider the following set of Pauli strings

$$P_\alpha \begin{cases} XXZ \\ YYZ \\ YZZ \end{cases} \Rightarrow P_\alpha^{0,1} \begin{cases} XX \\ YY \\ YZ \end{cases} \Rightarrow C_{01} = C_{10} = 2.$$

In this example $C_{10} = C_{01} = 2$ because XX and YY can be simultaneously measured with a Bell measurement and XX and YZ with an Ω^X measurement (see Supp. Material A 2). However, YY and YZ cannot be measured simultaneously, i.e., $[YY, YZ] \neq 0$.

With the compatibility matrix, we can tackle the *processor embedding* problem by trying to maximize the number of potential entangled measurements between neighboring qubits. We formulate this problem in terms of graphs, as shown in Fig. 1. Let $G' := (V', E', C)$ be the

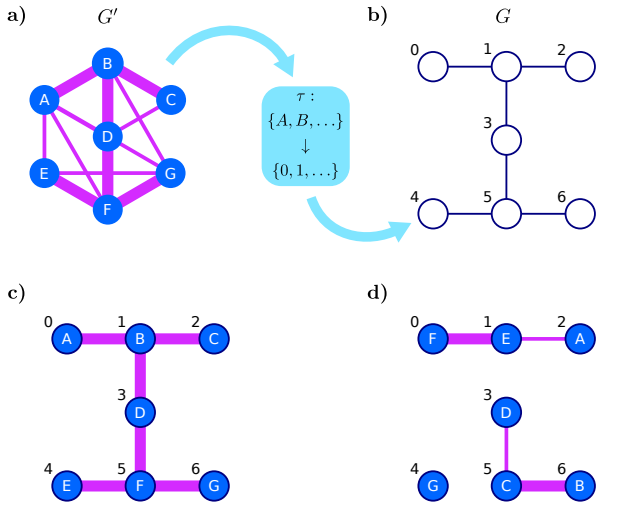


Figure 1. Schematic representation of the *processor embedding* problem. **a)** Compatibility graph of theoretical qubits, G'. Given a set of Pauli strings, the width of the edges $\{i, j\}$ of this graph represents the number of entangled measurements that can be established between qubits i and j, C_{ij}. **b)** Connectivity graph of the physical qubits of the `ibmq_jakarta` device, G. **c)** and **d)** are good and bad processor embedding maps τ, respectively.

weighted graph defined by the vertices $V' := \{1, \ldots, N\}$, where N is the number of theoretical qubits, the edges $E' := V' \times V'$ and the weights C_{ij}. In this way, G' represents the compatibility between the theoretical qubits (Fig. 1 a)). Let $G := (V, E)$ be the graph defined by the vertices $V := \{1, \ldots, M\}$, with $M \geq N$ the number of physical qubits and the edges $E := \{(i, j) \in V \times V :$ qubits i and j that are physically connected$\}$. In this way, G represents the topology of the chip (Fig. 1 b)). The objective is to find the map $\tau : \{1, \ldots, N\} \to \{1, \ldots, M\}$ that maximizes

$$\omega(\tau) = \sum_{\{(i,j) \in V' \times V' : (\tau(i), \tau(j)) \in E\}} C_{ij}. \tag{2}$$

Note that $\omega(\tau)$ is the total number of compatibilities between theoretical qubits once they are mapped to physical qubits through τ. For the example shown in Fig. 1 a), where the thin edges correspond to $C_{ij} = 1$ and the wide ones to $C_{ij} = 2$, the optimal processor embedding map (Fig. 1 c)) results in $\omega(\tau) = 12$, while a bad mapping (Fig. 1 d)) gives a lower value $\omega(\tau) = 6$.

The problem of finding the best τ is what we will refer to by the *processor embedding* problem, which we know is NP-Hard, and hence there is no efficient algorithm for the general case. There are several ways to prove that the *processor embedding* problem is NP-Hard. For example, the max-cliqué problem, which is itself NP-Hard [51], can be regarded as a simplification of the processor embedding problem so that all nondiagonal entries of C are positive and equal. Additionally, the k-densest subgraph problem is NP-Complete and can also be seen as a particular instance of the processor embedding problem [52, 53]. Efficient algorithms could be found for the physically relevant instances of the processor embedding problem if one finds some kind of structure within them. For now, we propose two heuristic algorithms (see Supp. Material B 2) to construct a map τ. The first is the *order disconnected* map (Algorithm 4). Here, each pair of theoretical qubits (i, j) with the highest entries C_{ij} is assigned to physically connected qubits. The second alternative is the *order connected* map (Algorithm 5). This algorithm does the same as the previous, but it additionally ensures that the graph $\tau(G')$ is a connected sub-graph of G. The third alternative is the *naive* map. This is simply the trivial map $\tau(i) = i$ for $i \in \{1, \ldots, N\}$, and it allows us to benchmark our *order connected* and *order disconnected* maps.

Once the map τ has been chosen one can propose an order for the loops of our grouping algorithm (Algorithm 1), both in the qubits and in the measurements order. Using the τ-compatibility matrix C^τ, we define

$$C^\tau_{ij} := \begin{cases} C_{ij} & \text{if } (\tau(i), \tau(j)) \in E \\ 0 & \text{if } (\tau(i), \tau(j)) \notin E. \end{cases} \tag{3}$$

Let CQ^τ be the N-vector whose i entry is given by

$$CQ^\tau_i := C\mathcal{X}_i + C\mathcal{Y}_i + C\mathcal{Z}_i + \sum_{j \neq i} C^\tau_{ij}, \tag{4}$$

where $C\mathcal{X}_i$, $C\mathcal{Y}_i$, and $C\mathcal{Z}_i$ are the numbers of compatibilities involving the qubit i through measurements $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively. In this way, CQ^τ_i is the number of compatibilities involving the qubit i, since τ has been chosen as the embedding map of the processor. Now, it is natural to propose the following order for the qubits: *run through the qubits in descending order of* CQ^τ.

Let CM^τ be the set of 9 tuples of the form

$$\begin{aligned} CM^\tau := \{&(\mathcal{X}, C\mathcal{X}), (\mathcal{Y}, C\mathcal{Y}), (\mathcal{Z}, C\mathcal{Z}), \\ &(\text{Bell}, C^\tau\text{Bell}), (\chi, C^\tau\chi), (\tilde{\chi}, C^\tau\tilde{\chi}), (\Omega^X, C^\tau\Omega^X), \\ &(\Omega^Y, C^\tau\Omega^Y), (\Omega^Z, C^\tau\Omega^Z)\}, \end{aligned} \tag{5}$$

where C^τBell is the number of compatibilities due to Bell measurement once τ has been chosen as the processor embedding map. $C^\tau\chi$, $C^\tau\tilde{\chi}$, $C^\tau\Omega^X$, $C^\tau\Omega^Y$ and $C^\tau\Omega^Z$ are defined analogously for the other entangled measurements (see Supp. Material). $C\mathcal{X}$, $C\mathcal{Y}$ and $C\mathcal{Z}$ are the numbers of compatibilities by the measurements $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ (these numbers do not depend on τ because these measurements are separable). Now, it is natural to propose the following order for the measurements: *run through the measurements in descending order of* CM^τ. The order in which Algorithm 1 runs along the Pauli strings is independent of τ. It is chosen by the Pauli graph, visiting the Pauli strings in descending order with respect to their degree in this graph, similarly to what is done in the largest degree-first coloring algorithm (LDFC) [54].

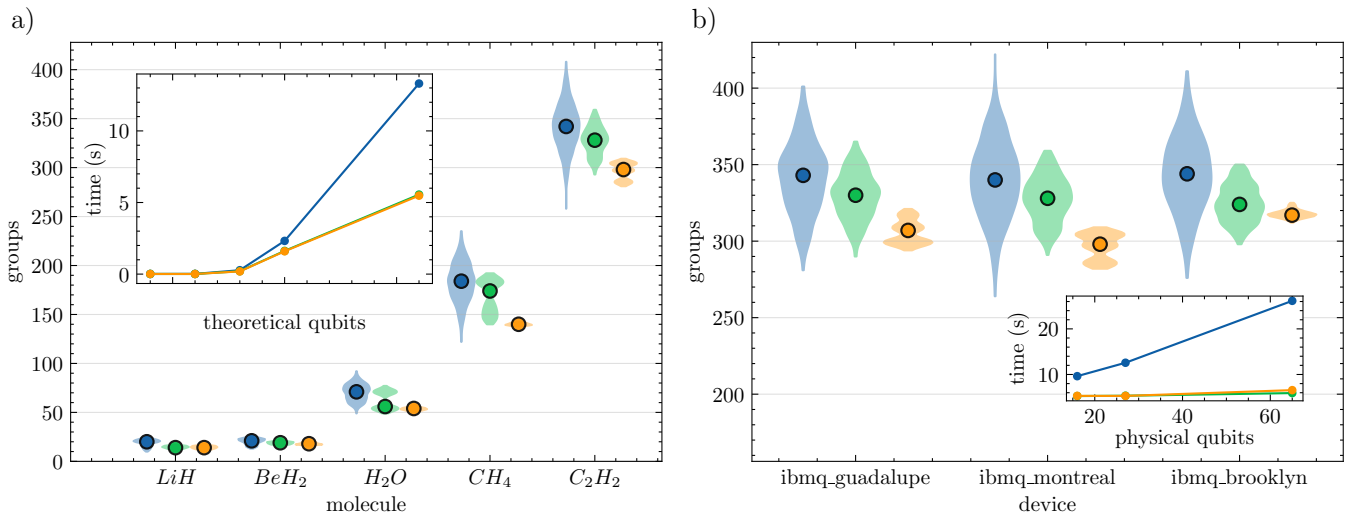


Figure 2. Results for grouping algorithms. The violins correspond to the distributions on the number of groups after running a Monte Carlo method over the order of the Pauli strings and the order of the theoretical qubits for different **a)** molecules, and **b)** devices. Blue violins correspond to the naive algorithm (Section II A), green violins correspond to the order disconnected algorithm (see Supp. Material, Algorithm 4), and orange violins to the order connected algorithm (see Supp. Material, Algorithm 5). The colored dots represent the mean value in the number of groups in each case. The insets show the average execution time of the grouping algorithm. In **a)** the algorithms group according to HEEM with the connectivity of *ibmq_ montreal*. In **b)** the number of groups corresponds to the C_2H_2 molecule.

Finally, we must emphasize that the *processor embedding* problem is not only relevant in choosing the best orders for the loops of our grouping algorithm (Algorithm 1), but also affects the number of groups obtained. For example, suppose that we have $N = M = 3$, two Pauli strings XXZ and ZXX and $E = \{(0,1),(1,2)\}$. If we choose $\tau: 0 \to 0,\ 1 \to 1,\ 2 \to 2$ we need to measure both strings separately, but choosing $\tilde{\tau}: 0 \to 1,\ 1 \to 0,\ 2 \to 2$ a single measurement suffices. In addition, the processor embedding problem is crucial to reduce the error caused by gates in NISQ computers, as a good choice of τ would reduce the number of CNOTs used in the circuit, which is a relevant task in the physical layout problem [55]. In fact, if we had to apply a theoretical CNOT in our algorithm between qubits i and j that are mapped to physical qubits whose distance in the chip is D, then in practice we would need to apply $\mathcal{O}(D)$ CNOTs between physically connected qubits, so reducing D using an appropriate τ is essential. Also, in a real device, the accuracy of the CNOT gates between connected qubits will vary depending on the chosen qubits. If one takes into account these other tasks and not only the grouping, the weights of G' should vary: it should not only depend on the compatibility matrix, but also on those other features. This extension will be addressed in future work.

III. RESULTS

We begin by comparing the three different HEEM grouping algorithms that we have proposed: the naive, the order disconnected, and the order connected methods. The reasons that led us to develop the order connected and order disconnected methods were improving the naive method and reducing its dependence on the order followed in its loops. To verify the enhancements, we studied the statistical behavior of the three algorithms with a Monte Carlo method that randomly permutes the order of the theoretical qubits and of the Pauli strings used as input for the algorithms. Indeed, Figure 2 a) shows that the order connected and order disconnected algorithms are better on average than the naive algorithm and that they have less dispersion, reducing their dependence on the orders. One might note that the absolute minimum number of groups is obtained with the naive algorithm, however, this is not relevant in practice since one would like to run the algorithm just once. It is also remarkable that the order-connected algorithm is the one that performs best by both having the lowest average value and the smallest dispersion, which means that this algorithm explores a region of the space of HEEMs that is more densely packed with good solutions.

In terms of time execution, the order-connected and order-disconnected algorithms are much faster than the naive one. The time complexity of the order-connected and disconnected algorithms is $\mathcal{O}(N^5)$ on the number of qubits in the Hamiltonian, versus the time complexity of $\mathcal{O}(N^{5.5})$ needed for a naive grouping. Figure 2 b) compares the performance of these three algorithms with three different processor architectures. It shows that the execution time of the non-naive algorithms does not depend on the architecture in contrast to the naive one does. This result is important for the practical application of the proposed algorithms on quantum devices with

	Qubits	No grouping	Grouping			CNOTs		Relative error (%)		
			TPB	EM	HEEM	EM	HEEM	TPB	EM	HEEM
H_2	2	5	2	2	2	1	1	2.9 ± 0.3	2.5 ± 0.2	2.4 ± 0.3
LiH	4	100	25	11	10	8	8	0 ± 1	0 ± 1	0.2 ± 0.9
BeH_2	6	95	24	15	13	74	18	-	-	-
H_2O	8	444	93	51	47	563	80	7 ± 1	9 ± 1	3.2 ± 0.8
CH_4	10	1181	246	113	117	2677	224	15 ± 2	11 ± 2	6 ± 3
C_2H_2	16	1884	457	189	258	8969	433	16 ± 2	19 ± 2	12 ± 3
CH_3OH	22	9257	2225	682	1503	9830	2770	25 ± 5	31 ± 3	20 ± 4
C_2H_6	26	8919	2069	758	1529	55809	2873	34 ± 7	40 ± 3	22 ± 4

Table I. Number of groups, number of CNOTs, and relative error of the energy evaluation for some molecules using different grouping strategies. TPB groups have been obtained using the LDFC algorithm. EM groups are those proposed by [46]. HEEM groups have been obtained using algorithm 1 assuming that the physical qubits have the connectivity of the device *ibmq_montreal*. Relative errors were obtained by simulations considering the noise model of the device *ibmq_montreal*, initializing the system in the $|0\rangle^{\otimes N}$ state. We omit the relative error of BeH_2 because the expected energy is equal to zero. Each simulation has a total of 2^{14} shots evenly distributed across all measurements in each grouping. The errors represent the standard deviation for a total of 25 simulations.

a large number of qubits. The opposite situation occurs for the grouping configurations: the naive algorithm does not differentiate between processors while the other two do. Thus, only the non-naive algorithms have the desired behavior with respect to different architectures: they are always quickly executed and can take advantage of the topology of the chip.

Let us now confront our grouping methods with the previous ones. Table I shows the number of groups obtained from the Pauli strings of molecules of sizes between 2 and 26 qubits using different methods: TPB grouping [41], EM grouping [46], and HEEM grouping with the connectivity of the quantum device *ibmq_montreal*. The results of the HEEM grouping correspond to the minimum number of groups obtained with our Monte Carlo study. Table I also includes the number of CNOTs required to measure the groups. On the one hand, the HEEM grouping significantly outperforms TPB in terms of the number of groups. However, the EM grouping has a smaller number of groups than the HEEM but uses a number of CNOTs that grows much more rapidly with the size of the molecules. The number of groups with HEEM scales as $\mathcal{O}(N^{2.45})$ with the number of qubits (the asymptotic scaling values for other magnitudes can be found in Supp. Material C). Furthermore, HEEM circuits involve only one layer of entangling gates, while the depth of EM circuits depends on the distance between the qubits to be connected.

Given that both the number of CNOTs, which translates into experimental error, and the number of groups, which translates into statistical error for fixed total number of shots, are relevant for the final accuracy of the method, the results discussed so far do not completely conclude which is the best grouping method. In Table I we show the relative error for the energy evaluation of the state $|\psi\rangle = |0\rangle^{\otimes N}$ for different molecule Hamiltonians and grouping methods. We consider this state because it is prepared with high quality since it does not have entanglement, so in the simulations only appear the error due to the energy evaluation. We set the total number of shots to 2^{14} for all molecules and methods, and these shots are divided equally between all the circuits of each scenario. With TPB grouping, no entangled measurements are performed, and hence there are no errors due to two-qubit gates. However, the total number of circuits to compute the energy is the largest, and consequently TPB has the lowest number of shots per circuit and the highest statistical error. With EM, the scenario is the opposite, as this method has the lowest statistical error, but it has the highest experimental error because it uses a large number of CNOT gates to connect distant qubits. HEEM takes the best of both methods, as it uses fewer groups than TPB, reducing the statistical error, and it does not connect distant qubits, reducing the experimental error. Due to this reason, as can be seen in Table I, HEEM provides the lowest relative error for all of the studied molecules.

Finally, to study the performance of the HEEM grouping in a practical scenario, we have run noisy simulations and an NISQ experiment of the VQE for the H_2O molecule at its bond distance. We freeze the core and remove an unoccupied orbital to reduce the number of theoretical qubits to 8. Figure 3 a) shows the mean energy per iteration of 240 instances of noisy simulations of VQE using TPB, EM, and HEEM. They were carried out using the noise model and the connectivity of *ibmq_montreal*, provided by IBM-Q [56]. The variational form is composed of 2 layers of local gates and 1 layer of CNOT between the physically connected qubits. The groups used for HEEM correspond to the best groups obtained among the three proposed algorithms. We can see that the grouping scheme with better performance is HEEM, achieving energies below -12.0 Ha after 200 iterations, while neither TPB nor EM reach this value in 300 iterations. The inset of Fig. 3 a) shows the mean en-

ergy in terms of the number of circuits. We can see that the HEEM grouping achieves an energy below −12.0 Ha with about 1200 circuits, while TPB and EM only attain −11.6 Ha and −11.2 Ha, respectively. This means that HEEM allows us to obtain the same results as the previous approaches but with fewer circuits. Figure 3 b) shows the experimental implementation of VQE for the H_2O with TPB, EM, and HEEM. The experiment was carried out on the first 8 qubits of the device *ibmq_guadalupe*. This is a quantum device of 16-qubits, 32 of quantum volume, and 1.245×10^{-2} of average error in the CNOT gates. We can see that the experimental results agree with what was expected from the simulations; that is, HEEM delivers an energy better than that of TPB and EM at the same number of iterations and using the same number of circuits. The time it took for each experiment was 3.8 hours with TPB, 4.4 hours with EM, and 2.7 hours with HEEM. Thus, HEEM provides a speed-up with respect to the other grouping approaches in real hardware.

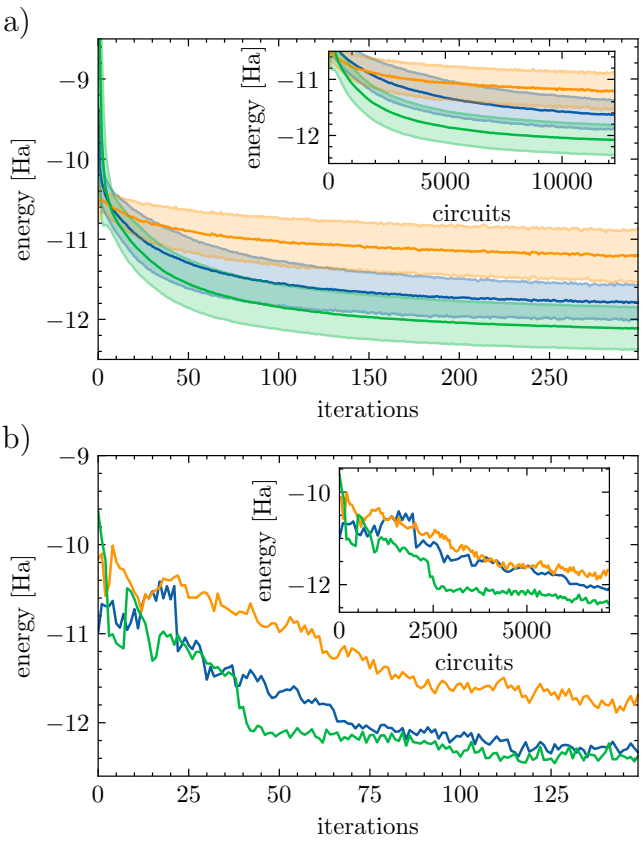


Figure 3. Simulation a), and experimental implementation b), of the VQE for the 8-qubits H_2O Hamiltonian, with a distance of $d = 0.96$Å between hydrogen and oxygen atoms, for TPB in blue, EM in orange, and HEEM in green lines. The simulations where performed considering the basis gate, noise model and connectivity of *ibmq_montreal*, while the real experiment is performed in *ibmq_guadalupe*. The classical optimizer used was SPSA with 300 iterations for the simulations and 150 iterations for the experiment. There are 2^{14} shots per circuit in both cases. a) Each point represents 240 independent instances of VQE where the solid lines correspond to the mean energy, and the shaded region to the standard deviation. The exact ground state energy is −13.9 Ha.

IV. CONCLUSIONS AND OUTLOOK

In the NISQ era, in order to achieve the quantum advantage with variational algorithms we need efficient techniques to evaluate the expected values of Hamiltonians. In this article, we have proposed to evaluate Hamiltonians with Hardware Efficient Entangled Measurements (HEEM), which allows one to evaluate simultaneously groups of Pauli strings, and employ entangled gates only between qubits that are physically connected on the device. This makes HEEM an efficient alternative for evaluating expected values and speeding up variational algorithms in NISQ devices. Since there are multiple ways to group a set of Pauli strings with HEEM, we have introduced three algorithms to carry out the grouping. The first is the *naive* HEEM, which does not optimize the order of the grouping loops. The other two are *order connected HEEM* and *the order disconnected HEEM*. They optimize the grouping loops by making use of the *processor embedding* problem, the problem of finding the optimal map of the theoretical qubits into the physical qubits.

We compare our methods and conclude that the order connected and order disconnected algorithms outperform the naive algorithm on average number of groups and time execution. Our methods improve previous works, using less groups than TPB [41], with a slight increase in the number of required CNOTs. Furthermore, HEEM requires less CNOTs than EM [46], using only a few more groups. The number of groups quantifies the statistical error, while the number of CNOTs quantifies the experimental error. Taking both sources of error into account, HEEM outperforms both TPB and EM. We have shown, with noisy simulations and experiments, that HEEM achieves better results with the same number of circuits. In addition, HEEM is faster than TPB and EM to be implemented both in an experiment and in a simulation for practical scenarios.

Several extensions can be incorporated into our proposal. The grouping of Pauli strings by HEEM can be improved by considering entanglement between a larger number of qubits. This would allow to further reduce the number of groups at the expense of more entanglement resources. The method can also be refined by including the error of the CNOT gates of the chip as weights in the graph that represents the connectivity of the chip. This would produce better results by ensuring that the majority of the entangling operations are performed over pairs of qubits with the least CNOT gate errors. Furthermore, we can implement ideas from previous methods in HEEM. Given that the number of groups and CNOTs

are proxies for the error of the measurement, we can explore the behaviour of HEEM under more elaborated metrics, such as the proposed in [47]. In the same article it is also suggested to sort the Pauli strings according to their weights in the Hamiltonian, which could improve HEEM in practical scenarios. Note that energy evaluation can also be improved with other methods, such as error mitigation techniques [57–59]. The combination of these methods with HEEM could provide an even more accurate estimation of observables on current devices.

HEEM grouping can be useful not only in variational algorithms, but also in any task that requires the evaluation of several Pauli strings, such as full quantum tomography [60, 61], compressed sensing [62, 63], reduced density matrix tomography [32, 33], classical shadows [29, 30], or direct fidelity estimation [64, 65]. The *processor embedding* problem also has applications in other branches of quantum information, such as the physical layout problem, and even in the industry.

The code to reproduce the algorithms and the figures of this article can be found in [66].

V. ACKNOWLEDGEMENTS

We thank Juan José García-Ripoll for valuable feedback on the manuscript. This work has been supported by funding from Spanish project PGC2018-094792-B-I00 (MCIU/AEI/FEDER, UE) and CAM/FEDER Project No. S2018/TCS-4342 (QUITEMAD-CM). G. F. P. and G. J. acknowledge support from the European Union's Horizon 2020 FET-Open project SuperQuLAN (899354). F.E.G. was supported by a Marie Skłodowska-Curie Action from the EC (COFUND grant no. 945045), and by the NWO Gravitation project NETWORKS (grant no. 024.002.003). D.F.F. acknowledges support from the FPU Program No. FPU20/04762. L.P. was supported by ANID-PFCHA/DOCTORADO-BECAS-CHILE/2019-77220027. The authors thank the IBM Quantum Team for making multiple devices available to the CSIC-IBM Quantum Hub via the IBM Quantum Experience. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

[1] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).
[2] I. G. Ryabinkin, S. N. Genin, and A. F. Izmaylov, Constrained variational quantum eigensolver: Quantum computer search engine in the fock space, Journal of Chemical Theory and Computation **15**, 249 (2018).
[3] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz, Quantum Science and Technology **4**, 014008 (2018).
[4] I. G. Ryabinkin, T.-C. Yen, S. N. Genin, and A. F. Izmaylov, Qubit coupled cluster method: A systematic approach to quantum chemistry on a quantum computer, Journal of Chemical Theory and Computation **14**, 6317 (2018).
[5] W. Kirby, B. Fuller, C. Hadfield, and A. Mezzacapo, Second-quantized fermionic operators with polylogarithmic qubit and gate complexity, PRX Quantum **3**, 020351 (2022).
[6] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).
[7] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, The variational quantum eigensolver: a review of methods and best practices (2021), arXiv:2111.05176 [quant-ph].
[8] A. F. Izmaylov, T.-C. Yen, and I. G. Ryabinkin, Revising the measurement process in the variational quantum eigensolver: is it possible to reduce the number of separately measured operators?, Chemical Science **10**, 3746 (2019).
[9] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5** (2014).
[10] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), arXiv:1411.4028 [quant-ph].
[11] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, Quantum optimization using variational algorithms on near-term quantum devices, Quantum Science and Technology **3**, 030503 (2018).
[12] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature **549**, 242 (2017).
[13] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos, Quantum chemistry calculations on a trapped-ion quantum simulator, Phys. Rev. X **8**, 031022 (2018).
[14] Y. Nam, J. S. Chen, N. C. Pisenti, K. Wright, C. Delaney, D. Maslov, K. R. Brown, S. Allen, J. M. Amini, J. Apisdorf, K. M. Beck, A. Blinov, V. Chaplin, M. Chmielewski, C. Collins, S. Debnath, K. M. Hudek, A. M. Ducore, M. Keesan, S. M. Kreikemeier, J. Mizrahi, P. Solomon, M. Williams, J. D. Wong-Campos, D. Moehring, C. Monroe, and J. Kim, Ground-state energy estimation of the water molecule on a trapped-ion quantum computer, npj Quantum Information **6**, 33 (2020).
[15] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, Quantum computing for finance: State-

of-the-art and future prospects, IEEE Transactions on Quantum Engineering **1**, 1 (2020).
[16] P. Vikstål, M. Grönkvist, M. Svensson, M. Andersson, G. Johansson, and G. Ferrini, Applying the quantum approximate optimization algorithm to the tail-assignment problem, Physical Review Applied **14**, 1 (2020).
[17] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni, and B. Parney, Traffic flow optimization using a quantum annealer, Frontiers in ICT **4**, 29 (2017).
[18] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).
[19] M. Benedetti, E. Grant, L. Wossnig, and S. Severini, Adversarial quantum circuit learning for pure state approximation, New Journal of Physics **21**, 043023 (2019).
[20] A. Patterson, H. Chen, L. Wossnig, S. Severini, D. Browne, and I. Rungger, Quantum state discrimination using noisy quantum neural networks, Physical Review Research **3**, 013063 (2021).
[21] H. Chen, L. Wossnig, S. Severini, H. Neven, and M. Mohseni, Universal discriminative quantum neural networks, Quantum Machine Intelligence **3**, 1 (2021).
[22] X. Wang, Z. Song, and Y. Wang, Variational quantum singular value decomposition, Quantum **5**, 483 (2021).
[23] K. Wang, Z. Song, X. Zhao, Z. Wang, and X. Wang, Detecting and quantifying entanglement on near-term quantum devices (2020), arXiv:2012.14311 [quant-ph].
[24] A. D. Muñoz-Moller, L. Pereira, L. Zambrano, J. Cortés-Vega, and A. Delgado, Variational determination of multi-qubit geometrical entanglement in nisq computers (2021), arXiv:2110.03709 [quant-ph].
[25] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Theory of variational quantum simulation, Quantum **3**, 191 (2019).
[26] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, Variational quantum algorithms for discovering hamiltonian spectra, Phys. Rev. A **99**, 062304 (2019).
[27] P. García-Molina, J. Rodríguez-Mediavilla, and J. J. García-Ripoll, Quantum fourier analysis for multivariate functions and applications to a class of schrödinger-type partial differential equations, Physical Review A **105**, 012433 (2022).
[28] P. Gokhale, O. Angiuli, Y. Ding, K. Gui, T. Tomesh, M. Suchara, M. Martonosi, and F. T. Chong, Minimizing state preparations in variational quantum eigensolver by partitioning into commuting families (2019), arXiv:1907.13623 [quant-ph].
[29] H. Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nature Physics **16**, 1050 (2020).
[30] S. Chen, W. Yu, P. Zeng, and S. T. Flammia, Robust shadow estimation, PRX Quantum **2**, 030348 (2021).
[31] S. Aaronson, Shadow tomography of quantum states, SIAM Journal on Computing **49**, STOC18 (2020).
[32] J. Cotler and F. Wilczek, Quantum overlapping tomography, Physical Review Letters **124**, 100401 (2020).
[33] X. Bonet-Monroig, R. Babbush, and T. E. O'Brien, Nearly optimal measurement scheduling for partial tomography of quantum states, Phys. Rev. X **10**, 031064 (2020).
[34] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, Restricted boltzmann machines in quantum physics, Nature Physics **15**, 887 (2019).
[35] G. Torlai, G. Mazzola, G. Carleo, and A. Mezzacapo, Precise measurement of quantum observables with neural-network estimators, Physical Review Research **2**, 022060 (2020).
[36] A. Shlosberg, A. J. Jena, P. Mukhopadhyay, J. F. Haase, F. Leditzky, and L. Dellantonio, Adaptive estimation of quantum observables (2021), arXiv:2110.15339 [quant-ph].
[37] H. J. Vallury, M. A. Jones, C. D. Hill, and L. C. L. Hollenberg, Quantum computed moments correction to variational estimates, Quantum **4**, 373 (2020).
[38] N. C. Rubin, R. Babbush, and J. McClean, Application of fermionic marginal constraints to hybrid quantum algorithms, New Journal of Physics **20**, 053020 (2018).
[39] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New Journal of Physics **18**, 023023 (2016).
[40] A. Arrasmith, L. Cincio, R. D. Somma, and P. J. Coles, Operator sampling for shot-frugal optimization in variational algorithms (2020), arXiv:2004.06252.
[41] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme, Tapering off qubits to simulate fermionic hamiltonians (2017), arXiv:1701.08213 [quant-ph].
[42] T.-C. Yen, V. Verteletskyi, and A. F. Izmaylov, Measuring all compatible operators in one series of single-qubit measurements using unitary transformations, Journal of Chemical Theory and Computation **16**, 2400 (2020).
[43] V. Verteletskyi, T.-C. Yen, and A. F. Izmaylov, Measurement optimization in the variational quantum eigensolver using a minimum clique cover, The Journal of Chemical Physics **152**, 124114 (2020).
[44] R. Kondo, Y. Sato, S. Koide, S. Kajita, and H. Takamatsu, Computationally Efficient Quantum Expectation with Extended Bell Measurements, Quantum **6**, 688 (2022).
[45] A. F. Izmaylov, T.-C. Yen, R. A. Lang, and V. Verteletskyi, Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method, Journal of Chemical Theory and Computation **16**, 190 (2020).
[46] I. Hamamura and T. Imamichi, Efficient evaluation of quantum observables using entangled measurements, npj Quantum Information **6**, 56 (2020).
[47] O. Crawford, B. van Straaten, D. Wang, T. Parks, E. Campbell, and S. Brierley, Efficient quantum measurement of pauli operators in the presence of finite sampling error, Quantum **5**, 385 (2021).
[48] A. Zhao, A. Tranter, W. M. Kirby, S. F. Ung, A. Miyake, and P. J. Love, Measurement reduction in variational quantum algorithms, Phys. Rev. A **101**, 062322 (2020).
[49] A. Jena, S. Genin, and M. Mosca, Pauli partitioning with respect to gate sets (2019), arXiv:1907.07859 [quant-ph].
[50] P. Formanowicz and K. Tanaś, A survey of graph coloring - its types, methods and applications, Foundations of Computing and Decision Sciences **37**, 223 (2012).
[51] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations* (Springer US, 1972) pp. 85–103.
[52] P. Manurangsi, Almost-polynomial ratio eth-hardness of approximating densest k-subgraph, in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017 (Association for Computing Machinery, New York, NY, USA, 2017) p. 954–961.

[53] R. Sotirov, On solving the densest k-subgraph problem on large graphs, Optimization Methods and Software **35**, 1160 (2020).

[54] C. Avanthay, A. Hertz, and N. Zufferey, A variable neighborhood search for graph coloring, European Journal of Operational Research **151**, 379 (2003).

[55] B. Tan and J. Cong, Optimal layout synthesis for quantum computing, in *Proceedings of the 39th International Conference on Computer-Aided Design* (ACM, 2020).

[56] IBM Quantum Computing (2022).

[57] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, Journal of the Physical Society of Japan **90**, 032001 (2021).

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

[59] Y. Li and S. C. Benjamin, Efficient variational quantum simulator incorporating active error minimization, Phys. Rev. X **7**, 021050 (2017).

[60] D. F. V. James, P. G. Kwiat, W. J. Munro, and A. G. White, Measurement of qubits, Phys. Rev. A **64**, 052312 (2001).

[61] H. Häffner, W. Hänsel, C. F. Roos, J. Benhelm, D. C. al kar, M. Chwalla, T. Körber, U. D. Rapol, M. Riebe, P. O. Schmidt, C. Becher, O. Gühne, W. Dür, and R. Blatt, Scalable multiparticle entanglement of trapped ions, Nature **438**, 643 (2005).

[62] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Quantum state tomography via compressed sensing, Phys. Rev. Lett. **105**, 150401 (2010).

[63] C. A. Riofrío, D. Gross, S. T. Flammia, T. Monz, D. Nigg, R. Blatt, and J. Eisert, Experimental quantum compressed sensing for a seven-qubit system, Nature Communications **8**, 15305 (2017).

[64] S. T. Flammia and Y.-K. Liu, Direct fidelity estimation from few pauli measurements, Phys. Rev. Lett. **106**, 230501 (2011).

[65] M. P. da Silva, O. Landon-Cardinal, and D. Poulin, Practical characterization of quantum devices without tomography, Phys. Rev. Lett. **107**, 210404 (2011).

[66] F. Escudero, D. Fernández-Fernández, G. Jaumà, G. F. Peñas, and L. Pereira, Hardware efficient variational quantum eigensolver with entangled measurements: Code repository (2022).

Supplemental Material

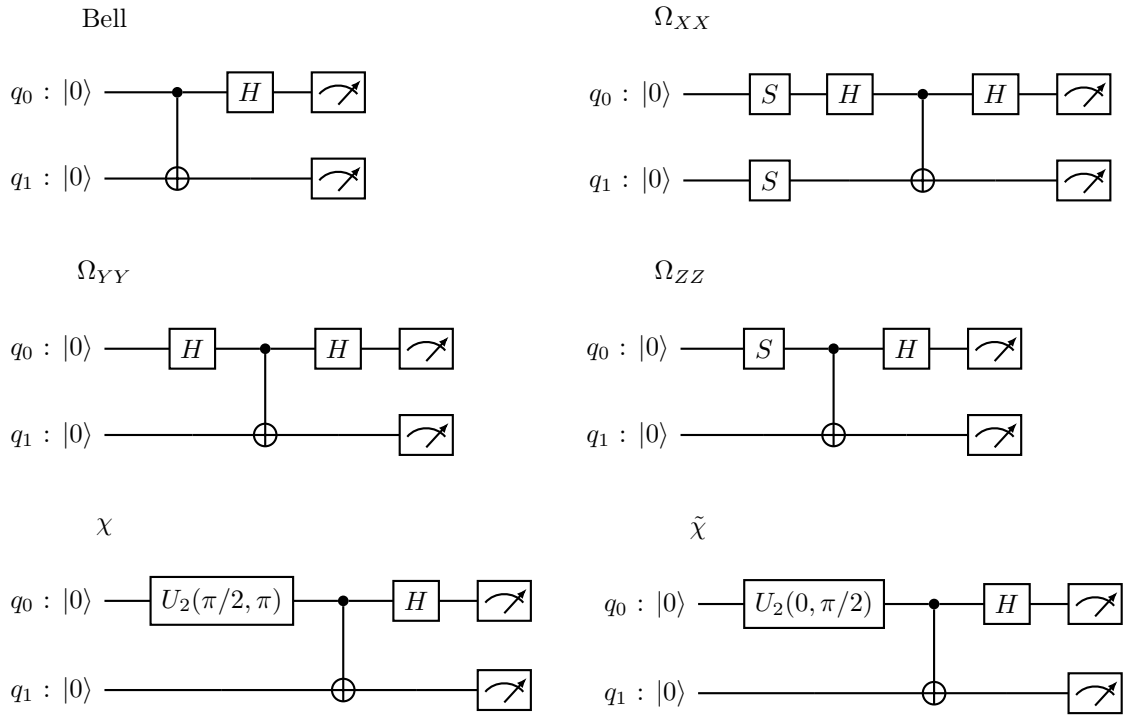
Hardware-efficient entangled measurements for variational quantum algorithms

Francisco Escudero,[1,2] David Fernández-Fernández,[1,3], Gabriel Jaumà,[1] Guillermo F. Peñas,[1] Luciano Pereira[1]

[1] *Instituto de Física Fundamental, IFF-CSIC, Calle Serrano 113b, 28006 Madrid, Spain*
[2] *Networks, Qusoft and CWI, Amsterdam, Netherlands.*
[3] *Instituto de Ciencia de Materiales de Madrid, ICMM-CSIC, 28049 Madrid, Spain*

Appendix A: Entangled measurements

We have identified the relevant two-qubit entangled measurements and explicitly constructed the circuits to perform each of them. They are the following ones:



1. Computing the expected values of Hamiltonians with a certain grouping

In this section, we explain how to obtain the expected value of a Hamiltonian once we have grouped its Pauli strings. Let $H = \sum_\alpha h_\alpha P_\alpha$ be a multi-qubit Hamiltonian and suppose that the m first Pauli strings are compatible with a single HEEM measurement into the basis $\mathcal{B} = \mathcal{B}_1 \otimes \cdots \otimes \mathcal{B}_t$, with B_i single or two qubit bases with $i = 1, \ldots, t \leq m$. This means that the Pauli strings P_α with $\alpha = 1, \ldots, m$ are diagonal in the basis $\mathcal{B}$. Let be $\vec{W}_P = \text{Diag}_{\mathcal{B}}\{P\}$ the vector with the diagonal entries of the Pauli string P on the basis of $\mathcal{B}$. The elements of $\vec{W}_P$ correspond to the eigenvalues of P, each of them repeated by its multiplicity. Thus, if $\vec{\mathcal{P}}$ is the probability distribution of a measurement on the basis $\mathcal{B}$, the expected value of the compatible Pauli strings is given by

$$\left\langle \sum_{\alpha=1}^{m} h_\alpha P_\alpha \right\rangle = \left(\sum_{\alpha=1}^{m} h_\alpha \vec{W}_{P_\alpha}, \vec{\mathcal{P}} \right), \tag{A1}$$

where $(\cdot,\cdot)$ is the inner product. In order to illustrate the procedure we are going to evaluate the energy of a simple Hamiltonian consisting only of two Pauli strings,

$$H = 2IZY + 4ZXZ. \tag{A2}$$

We have to check in Table II on which basis, if possible, these two strings can be measured together. After a careful look, we realize that the operators I and Z of the first qubits are compatible with $\mathcal{Z}$, and that the operators XY and XZ of the second and third qubits are compatibles with $\tilde{\chi}$. The basis $\mathcal{Z}$ allows us to diagonalize the operators $I = +|0\rangle\langle 0| + |1\rangle\langle 1|$, and $Z = +|0\rangle\langle 0| - |1\rangle\langle 1|$, while the basis $\tilde{\chi}$ the 2-qubit operators

$$YX = -|\tilde{\chi}_0\rangle\langle\tilde{\chi}_0| + |\tilde{\chi}_1\rangle\langle\tilde{\chi}_1| + |\tilde{\chi}_2\rangle\langle\tilde{\chi}_2| - |\tilde{\chi}_3\rangle\langle\tilde{\chi}_3| \tag{A3}$$
$$ZY = +|\tilde{\chi}_0\rangle\langle\tilde{\chi}_0| + |\tilde{\chi}_1\rangle\langle\tilde{\chi}_1| - |\tilde{\chi}_2\rangle\langle\tilde{\chi}_2| - |\tilde{\chi}_3\rangle\langle\tilde{\chi}_3| \tag{A4}$$
$$XZ = +|\tilde{\chi}_0\rangle\langle\tilde{\chi}_0| - |\tilde{\chi}_1\rangle\langle\tilde{\chi}_1| + |\tilde{\chi}_2\rangle\langle\tilde{\chi}_2| - |\tilde{\chi}_3\rangle\langle\tilde{\chi}_3|. \tag{A5}$$

One last important thing that we need is the explicit dependence of the vectors of the $\tilde{\chi}$ basis with respect to the computational basis, to correctly relate the vector of weights with the vector of outcomes

$$|\tilde{\chi}_0\rangle = +i|00\rangle - |01\rangle + i|10\rangle + |11\rangle \tag{A6}$$
$$|\tilde{\chi}_1\rangle = +|00\rangle + i|01\rangle - |10\rangle + i|11\rangle \tag{A7}$$
$$|\tilde{\chi}_2\rangle = +i|00\rangle + |01\rangle + i|10\rangle - |11\rangle \tag{A8}$$
$$|\tilde{\chi}_3\rangle = -|00\rangle + i|01\rangle + |10\rangle + i|11\rangle. \tag{A9}$$

See subsection A 2 for explicit expressions of how to construct every entangled measurement and the diagonal representation of pairs of observables in all of these bases. Now we can tackle the problem of evaluating the energy of Eq. (A2) with a measurement in the basis $\mathcal{B} = \mathcal{Z}_1 \otimes \tilde{\chi}_{2,3}$, where the subscripts refer to the subspace spanned by such qubits. For the Pauli string IZY we have $\mathrm{Diag}_{\mathcal{Z}}\{I\} = [+1,+1]$ and $\mathrm{Diag}_{\tilde{\chi}}\{ZY\} = [+1,+1,-1,-1]$. The vector $\vec{W}_{IZY}$ is computed as the Kronecker product of the vectors $\mathrm{Diag}_{\mathcal{Z}}$ and $\mathrm{Diag}_{\tilde{\chi}}$. Thus,

$$h_{IZY}\vec{W}_{IZY} = 2 \times [+1,+1] \otimes [+1,+1,-1,-1] = [+2,+2,-2,-2,+2,+2,-2,-2]. \tag{A10}$$

Analogously for the string ZXZ, we have $\mathrm{Diag}_{\mathcal{Z}}\{Z\} = [+1,-1]$ and $\mathrm{Diag}_{\tilde{\chi}}\{XZ\} = [+1,-1,+1,-1]$. Then

$$h_{ZXZ}\vec{W}_{ZXZ} = 4 \times [+1,-1] \otimes [+1,-1,+1,-1] = [+4,-4,+4,-4,-4,+4,-4,+4]. \tag{A11}$$

Therefore, we have that

$$\sum_{\alpha=1}^{m} h_\alpha \vec{W}_{P_\alpha} = [+6,-2,+2,-6,-2,+6,-6,+2]. \tag{A12}$$

In summary, what our algorithm does once it has identified which Pauli strings can be measured together is to check with which measurement basis it can do so and in which order those basis vectors have to be taken into account. After sorting out all the plus and minus signs, all is left to do is to multiply by the weight in the Hamiltonian h_α, and finally plug in the actual result of the experiment with Eq. (A1).

	XX	YZ	ZY	YY	XZ	ZX	ZZ	XY	YX
XX	—	Ω^X	Ω^X	Bell	✖	✖	Bell	✖	✖
YZ		—	Ω^X	✖	✖	χ	✖	χ	✖
ZY			—	✖	$\tilde{\chi}$	✖	✖	✖	$\tilde{\chi}$
YY				—	Ω^Y	Ω^Y	Bell	✖	✖
XZ					—	Ω^Y	✖	✖	$\tilde{\chi}$
ZX						—	✖	χ	✖
ZZ							—	Ω^Z	Ω^Z
XY								—	Ω^Z
YX									—

Table II. Compatibility relation between two-qubits Pauli strings. The symbol ✖ indicates that the corresponding Pauli strings are not jointly measurable. The other boxes contain the entangled measurements that are compatible with the corresponding Pauli strings. See Appendix A 2 for the definitions of these measurements.

2. Jointly diagonalizable pairs in all entangled bases.

$Bell$:

$$|\Phi_0\rangle = |00\rangle + |11\rangle$$
$$|\Phi_1\rangle = |00\rangle - |11\rangle$$
$$|\Phi_2\rangle = |01\rangle + |10\rangle$$
$$|\Phi_3\rangle = |01\rangle - |10\rangle$$

Commuting pairs

$$XX = +|\Phi_0\rangle\langle\Phi_0| - |\Phi_1\rangle\langle\Phi_1| + |\Phi_2\rangle\langle\Phi_2| - |\Phi_3\rangle\langle\Phi_3|$$
$$YY = -|\Phi_0\rangle\langle\Phi_0| + |\Phi_1\rangle\langle\Phi_1| + |\Phi_2\rangle\langle\Phi_2| - |\Phi_3\rangle\langle\Phi_3|$$
$$ZZ = +|\phi_0\rangle\langle\Phi_0| + |\Phi_1\rangle\langle\Phi_1| - |\Phi_2\rangle\langle\Phi_2| - |\Phi_3\rangle\langle\Phi_3|$$

Ω^X :

$$|\Omega_0^X\rangle = +|00\rangle - i|01\rangle - i|10\rangle + |11\rangle$$
$$|\Omega_1^X\rangle = +|00\rangle - i|01\rangle + i|10\rangle - |11\rangle$$
$$|\Omega_2^X\rangle = +|00\rangle + i|01\rangle - i|10\rangle - |11\rangle$$
$$|\Omega_3^X\rangle = -|00\rangle - i|01\rangle - i|10\rangle - |11\rangle$$

Commuting pairs

$$YZ = -\left|\Omega_0^X\right\rangle\left\langle\Omega_0^X\right| + \left|\Omega_1^Y\right\rangle\left\langle\Omega_1^X\right| - \left|\Omega_2^X\right\rangle\left\langle\Omega_2^X\right| + \left|\Omega_3^X\right\rangle\left\langle\Omega_3^X\right|$$
$$XX = +\left|\Omega_0^X\right\rangle\left\langle\Omega_0^X\right| - \left|\Omega_1^X\right\rangle\left\langle\Omega_1^X\right| - \left|\Omega_2^X\right\rangle\left\langle\Omega_2^X\right| + \left|\Omega_3^X\right\rangle\left\langle\Omega_3^X\right|$$
$$ZY = -\left|\Omega_0^X\right\rangle\left\langle\Omega_0^X\right| - \left|\Omega_1^X\right\rangle\left\langle\Omega_1^X\right| + \left|\Omega_2^X\right\rangle\left\langle\Omega_2^X\right| + \left|\Omega_3^X\right\rangle\left\langle\Omega_3^X\right|$$

Ω^Y :

$$|\Omega_0^Y\rangle = +|00\rangle + |01\rangle + |10\rangle - |11\rangle$$
$$|\Omega_1^Y\rangle = +|00\rangle + |01\rangle - |10\rangle + |11\rangle$$
$$|\Omega_2^Y\rangle = +|00\rangle - |01\rangle + |10\rangle + |11\rangle$$
$$|\Omega_3^Y\rangle = -|00\rangle + |01\rangle + |10\rangle + |11\rangle$$

Commuting pairs

$$XZ = +\left|\Omega_0^Y\right\rangle\left\langle\Omega_0^Y\right| - \left|\Omega_1^Y\right\rangle\left\langle\Omega_1^Y\right| + \left|\Omega_2^Y\right\rangle\left\langle\Omega_2^Y\right| - \left|\Omega_3^Y\right\rangle\left\langle\Omega_3^Y\right|$$
$$YY = +\left|\Omega_0^Y\right\rangle\left\langle\Omega_0^Y\right| - \left|\Omega_1^Y\right\rangle\left\langle\Omega_1^Y\right| - \left|\Omega_2^Y\right\rangle\left\langle\Omega_2^Y\right| + \left|\Omega_3^Y\right\rangle\left\langle\Omega_3^Y\right|$$
$$ZX = +\left|\Omega_0^Y\right\rangle\left\langle\Omega_0^Y\right| + \left|\Omega_1^Y\right\rangle\left\langle\Omega_1^Y\right| - \left|\Omega_2^Y\right\rangle\left\langle\Omega_2^Y\right| - \left|\Omega_3^Y\right\rangle\left\langle\Omega_3^Y\right|$$

Ω^Z :

$$|\Omega_0^Z\rangle = +|00\rangle - i|11\rangle$$
$$|\Omega_1^Z\rangle = +|00\rangle + i|11\rangle$$
$$|\Omega_2^Z\rangle = +|01\rangle + i|10\rangle$$
$$|\Omega_3^Z\rangle = +|00\rangle - i|10\rangle$$

Commuting pairs

$$XY = -\left|\Omega_0^Z\right\rangle\left\langle\Omega_0^Z\right| + \left|\Omega_1^Z\right\rangle\left\langle\Omega_1^Z\right| - \left|\Omega_2^Z\right\rangle\left\langle\Omega_2^Z\right| + \left|\Omega_3^Z\right\rangle\left\langle\Omega_3^Z\right|$$
$$YX = -\left|\Omega_0^Z\right\rangle\left\langle\Omega_0^Z\right| + \left|\Omega_1^Z\right\rangle\left\langle\Omega_1^Z\right| + \left|\Omega_2^Z\right\rangle\left\langle\Omega_2^Z\right| - \left|\Omega_3^Z\right\rangle\left\langle\Omega_3^Z\right|$$
$$ZZ = +\left|\Omega_0^Z\right\rangle\left\langle\Omega_0^Z\right| + \left|\Omega_1^Z\right\rangle\left\langle\Omega_1^Z\right| - \left|\Omega_2^Z\right\rangle\left\langle\Omega_2^Z\right| - \left|\Omega_3^Z\right\rangle\left\langle\Omega_3^Z\right|$$

χ :

$$|\chi_0\rangle = -|00\rangle + |01\rangle - i|10\rangle + i|11\rangle$$
$$|\chi_1\rangle = +|00\rangle + |01\rangle + i|10\rangle - i|11\rangle$$
$$|\chi_2\rangle = +|00\rangle - |01\rangle + i|10\rangle + i|11\rangle$$
$$|\chi_3\rangle = -|00\rangle + |01\rangle + i|10\rangle + i|11\rangle$$

Commuting pairs

$$XY = +|\chi_0\rangle\langle\chi_0| - |\chi_1\rangle\langle\chi_1| + |\chi_2\rangle\langle\chi_2| - |\chi_3\rangle\langle\chi_3|$$
$$YZ = -|\chi_0\rangle\langle\chi_0| + |\chi_1\rangle\langle\chi_1| + |\chi_2\rangle\langle\chi_2| - |\chi_3\rangle\langle\chi_3|$$
$$ZX = +|\chi_0\rangle\langle\chi_0| + |\chi_1\rangle\langle\chi_1| - |\chi_2\rangle\langle\chi_2| - |\chi_3\rangle\langle\chi_3|$$

$\tilde{\chi}$:

$$|\tilde{\chi}_0\rangle = +i|00\rangle - |01\rangle + i|10\rangle + |11\rangle$$
$$|\tilde{\chi}_1\rangle = +|00\rangle + i|01\rangle - |10\rangle + i|11\rangle$$
$$|\tilde{\chi}_2\rangle = +i|00\rangle + |01\rangle + i|10\rangle - |11\rangle$$
$$|\tilde{\chi}_3\rangle = -|00\rangle + i|01\rangle + |10\rangle + i|11\rangle$$

Commuting pairs

$$YX = -|\tilde{\chi}_0\rangle\langle\tilde{\chi}_0| + |\tilde{\chi}_1\rangle\langle\tilde{\chi}_1| + |\tilde{\chi}_2\rangle\langle\tilde{\chi}_2| - |\tilde{\chi}_3\rangle\langle\tilde{\chi}_3|$$
$$ZY = +|\tilde{\chi}_0\rangle\langle\tilde{\chi}_0| + |\tilde{\chi}_1\rangle\langle\tilde{\chi}_1| - |\tilde{\chi}_2\rangle\langle\tilde{\chi}_2| - |\tilde{\chi}_3\rangle\langle\tilde{\chi}_3|$$
$$XZ = +|\tilde{\chi}_0\rangle\langle\tilde{\chi}_0| - |\tilde{\chi}_1\rangle\langle\tilde{\chi}_1| + |\tilde{\chi}_2\rangle\langle\tilde{\chi}_2| - |\tilde{\chi}_3\rangle\langle\tilde{\chi}_3|$$

Appendix B: Algorithms

In this appendix the heuristic algorithms announced in the main text are provided and motivated in more detail. All of them have been implemented in a library compatible with QISKIT [66].

1. Taking chip's connectivity into account

Herewith we expose the modification made to the algorithm in [46] needed to take into account chip's connectivity. This modification is the inclusion of the line 8 in the Algorithm 2. In addition, we explain why this algorithm depends on the orders of the iterative elements, as announced in section II of the main text. The line 8 of Algorithm 2 shows that this algorithm depends on the processor embedding map chosen (in this version of the algorithm τ is implicitly chosen to be $\tau(i) = i$ for all $i \in \{0, \ldots, N-1\}$). The line 6 of Algorithm 2 shows the dependence on the order used to run through the assignable measurements. The line 7 of Algorithm 2 shows the dependence in the order of the qubits. Finally, the lines 4 and 9 of Algorithm 1 show that the algorithm depends on the order of the Pauli strings (in this version of the algorithm, the Pauli string order is chosen to be the largest degree-first). In the Subsection B 2 we provide heuristic algorithms to choose these orders adequately.

Algorithm 1: Greedy grouping taking chip's connectivity into account

Data: A set of n Pauli strings PS of N qubits, a set of measurements $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_l\}$ and a topology graph $G = (V, E)$

1 Build the Pauli graph of PS; let $\{v_i\}_{i=1}^n$ be its vertices
2 Initialize the measurements M
3 Initialize the groups Gr
4 **for** *i=1,...,n* **do**
5 **if** v_i *is not in any group* **then**
6 Initialize measurement M_i and assign M_i to v_i
7 Initialize the group Gr_i and add v_i to Gr_i
8 *# M_i stores a set of local measurements, indicating in which qubits they act, such that are compatible with all Pauli strings in group Gr_i*
9 **for** *j=i+1,...,n* **do**
10 **if** v_j *is not in any group* **then**
11 **if** v_i *and* v_j *are jointly measurable with respect to* M_i, $\mathcal{E}$ *and* G *(see algorithm 2)* **then**
12 Update M_i and add v_j to Gr_i
13 Append M_i to M and Gr_i to Gr
14 **return** M, Gr

Algorithm 2: Greedy measurement assignment taking chip's connectivity into account

Data: Two Pauli strings v_i and v_j, a set of measurements $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_l\}$, a topology graph $G = (V, E)$, and a current assignment of a measurement M_i

1 **if** *v_j is not compatible with M_i* **then**
2 **return** ***fail***
3 Let U be the indices of qubits without an assigned local measurement in M_i
4 Remove from U the indices where v_i and v_j coincide
5 **while** $U \neq \emptyset$ **do**
6 **for** $\varepsilon \in \mathcal{E}$ **do**
7 **for** $p \in$ *permutations of U of size equal to the number of qubits where ε acts* **do**
8 **if** *qubits of p are in E (only do if p has size bigger than 1)* **then**
9 **if** *v_i and v_j are compatible with ε in the qubits at position p* **then**
10 Update M_i with ε at position p
11 Remove the qubits of p from U
12 Go to line 5
13 **return** ***fail***
14 **return** ***success***, M_i

2. Processor embedding and order algorithms

In this section we provide heuristic algorithms to propose adequate orders to the iterative items of Algorithms 1 and 2. In particular, we provide a heuristic algorithm for the *processor embedding* problem. Algorithm 3 computes the compatibility matrix, and Algorithm 4 proposes a processor embedding map that aims to map the pair of theoretical qubits (i, j) with the highest entries C_{ij} to physically connected qubits. Algorithm 5 has the same final purpose as Algorithm 4, with the additional requirement of ensuring that the graph G' is connected once is mapped to G through τ, i.e., $\tau(G')$ is a connected subgraph of G. Although Algorithm 5 may look like an arbitrary modification of Algorithm 4, the results show that performance improves when imposing that $\tau(G')$ is connected (see Figure 2 of the main text). Finally, Algorithm 6 outputs the τ-compatibility matrix C^τ and the dictionaries CM^τ and CQ^τ.

Algorithm 3: Construction of compatibility matrix

Data: A set of n Pauli strings PS of N qubits

1 Initialize C as an $N \times N$ matrix
2 **for** $i \in \{0, \ldots, N-1\}$ **do**
3 **for** $j \in \{i+1, \ldots, N-1\}$ **do**
4 $PS' = PS[:, [i, j]]$
5 *# $F(XX)$ is the number of factors X appearing in PS'*
6 $CM\text{Bell}_{ij} = \left\{[F(II) + F(XX) + F(YY) + F(ZZ)]^2 - [F(II) + F(XX) + F(YY) + F(ZZ)]\right\}/2$
7 $CM\chi_{ij} = \left\{[F(II) + F(XY) + F(YZ) + F(ZX)]^2 - [F(II) + F(XY) + F(YZ) + F(ZX)]\right\}/2$
8 .
9 .
10 .
11 $C[i, j] = C[j, i] = CM\text{Bell}_{ij} + CM\chi_{ij} + CM\tilde{\chi}_{ij} + CM\Omega^X_{ij} + CM\Omega^Y_{ij} + CM\Omega^Z_{ij}$
12 **return** C

Algorithm 4: Processor embedding map construction

Data: A compatibility matrix C and a topology graph $G = (V, E)$

Initialize a list $AQ = [\,]$; *# It stores the theoretical qubits with an assigned physical qubit.*
Initialize a list $\tau = [\,]$; *# It stores the information of the processor embedding map.*
while $AQ \neq \{1, \ldots, N-1\}$ **do**
    Choose (i_0, j_0) such that $C_{i_0,j_0} = \max_{ij} C_{ij}$
    if $i_0 \in AQ$ ***and*** $j_0 \in AQ$ **then**
        $C[j_0, i_0] = C[i_0, j_0] = \text{NaN}$
    else if $i_0 \in AQ$ ***or*** $j_0 \in AQ$ **then**
        #If $i_0 \in AQ$, then i_0 and j_0 cannot simultaneously belong to AQ. The same applies to j_0.
        Let J be not assigned neighbor of $\tau(i_0)$
        Append (j_0, J) to τ (so $\tau(j_0) = J$)
        Append j_0 to AQ
        $C[i_0, j_0] = C[j_0, i_0] = \text{NaN}$
        for $S \in$ *neighbors of* $\tau(j_0)$ **do**
            if S *is assigned to a theoretical qubit* **then**
                $E = E - (S, \tau(j_0))$
                if *degree*$(S) == 0$ **then**
                    s = theoretical qubit assigned to the physical qubit S
                    $C[s, :] = C[:, s] = \text{NaN}$
        if *degree*$(\tau(j_0)) == 0$ **then**
            $C[j_0, :] = C[:, j_0] = \text{NaN}$
        # degree$(\tau(i_0))$=0 means that the physical qubit correspondent to the theoretical qubit i_0 has no available neighbors (because we update the connectivity graph by deleting the edges corresponding to pairs of physical qubits that have been assigned). Setting $C[i_0, :]$ and $C[:, i_0]$ to NaN ensures that no pair containing i_0 will be chosen at the beginning of the while. By doing that we decrease the number of iterations inside the while.
    else
        if *there is a pair of unassigned and connected physical qubits* **then**
            Append (j_0, J) and (i_0, I) to τ (so $\tau(i_0) = I$ and $\tau(j_0) = J$)
            Append i_0 and j_0 to AQ
            $C[i_0, j_0] = C[j_0, i_0] = \text{NaN}$
            for $S \in$ *neighbors of* $\tau(i_0)$ **do**
                if S *is assigned to a theoretical qubit* **then**
                    $E = E - (S, \tau(i_0))$
                    if *degree*$(S) == 0$ **then**
                        s = theoretical qubit assigned to the physical qubit S
                        $C[s, :] = C[:, s] = \text{NaN}$
            for $S \in$ *neighbors of* $\tau(j_0)$ **do**
                if S *is assigned to a theoretical qubit* **then**
                    $E = E - (S, \tau(j_0))$
                    if *degree*$(S) == 0$ **then**
                        s = theoretical qubit assigned to the physical qubit S
                        $C[s, :] = C[:, s] = \text{NaN}$
            if *degree*$(\tau(i_0)) == 0$ **then**
                $C[i_0, :] = C[:, i_0] = \text{NaN}$
        else
            $C[i_0, j_0] = C[j_0, i_0] = \text{NaN}$
return τ

Algorithm 5: Processor embedding map construction, ensuring that $\tau(G')$ is connected

Data: A compatibility matrix C and a topology graph $G = (V, E)$
Initialize a list $AQ = [\,]$ # *It stores the theoretical qubits with an assigned physical qubit.*
Initialize a list $\tau = [\,]$ # *It stores the information about the processor embedding map.*
Choose (i_0, j_0) such that $C_{i_0 j_0} = \max_{ij} C_{ij}$
Let (I, J) be a pair of physically connected qubits
Append (j_0, J) and (i_0, I) to τ
Append i_0 and j_0 to AQ
$C[i_0, j_0] = C[j_0, i_0] = \text{NaN}$
$E = E - (\tau(i_0), \tau(j_0))$
if *degree*$(\tau(i_0)) == 0$ **then**
    $C[i_0, :] = C[:, i_0] = \text{NaN}$
if *degree*$(\tau(j_0)) == 0$ **then**
    $C[j_0, :] = C[:, j_0] = \text{NaN}$
while $AQ \neq \{1, \ldots, N-1\}$ **do**
    $C' = C[AQ, :]$
    Choose (i_0, j_0) such that $C'_{i_0, j_0} = \max_{ij} C'_{ij}$
    if $i_0 \in AQ$ ***and*** $j_0 \in AQ$ **then**
        $C[j_0, i_0] = C[i_0, j_0] = \text{NaN}$
    else if $i_0 \in AQ$ ***or*** $j_0 \in AQ$ **then**
        #If $i_0 \in AQ$, then i_0 and j_0 cannot simultaneously belong to AQ. Same for j_0.
        Let J be not assigned neighbor of $\tau(i_0)$
        Append (j_0, J) to τ (so $\tau(j_0) = J$)
        Append j_0 to AQ
        $C[i_0, j_0] = C[j_0, i_0] = \text{NaN}$
        for $S \in$ *neighbors of* $\tau(j_0)$ **do**
            if S *is assigned to a theoretical qubit* **then**
                $E = E - (S, \tau(j_0))$
                if *degree*$(S) == 0$ **then**
                    s = theoretical qubit assigned to the physical qubit S
                    $C[s, :] = C[:, s] = \text{NaN}$
        if *degree*$(\tau(j_0)) == 0$ **then**
            $C[j_0, :] = C[:, j_0] = \text{NaN}$
        # *degree$(\tau(i_0))$=0 means that the physical qubit correspondent to the theoretical qubit i_0 has no available neighbors (because we update the connectivity graph by deleting the edges corresponding to pairs of physical qubits that have been assigned). Setting $C[i_0, :]$ and $C[:, i_0]$ to -1 ensures that no pair containing i_0 will be chosen at the beginning of the while. By doing that we decrease the number of iterations inside the while.*
return τ

Algorithm 6: Construction of τ-compatibility matrix

Data: A set of n Pauli strings PS of N qubits, a processor embedding map τ and a topology graph $G = (V, E)$

Initialize C^τ as an $N \times N$ matrix
Initialize CM^τ as a dictionary
$CM^\tau = \{(\mathcal{X}, 0), (\mathcal{Y}, 0), (\mathcal{Z}, 0), (Bell, 0), (\chi, 0), (\tilde{\chi}, 0), (\Omega^X, 0), (\Omega^Y, 0), (\Omega^Z, 0)\}$
Initialize CQ^τ as a zero array of length N
for $i \in \{0, \ldots, N-1\}$ **do**
  for $j \in \{i+1, \ldots, N-1\}$ **do**
    if $(\tau(i), \tau(j)) \in E$ **then**
      $PS' = PS[:, [i, j]]$
      # $F(XX)$ is the number of factors X appearing in PS'
      $C^\tau \text{Bell}_{ij} = \{[F(II) + F(XX) + F(YY) + F(ZZ)]^2 - [F(II) + F(XX) + F(YY) + F(ZZ)]\}/2$
      $C^\tau \chi_{ij} = \left\{[F(II) + F(XY) + F(YZ) + F(ZX)]^2 - [F(II) + F(XY) + F(YZ) + F(ZX)]\right\}/2$
      .
      .
      .
      $C^\tau[i, j] = C^\tau[j, i] = C^\tau \text{Bell}_{ij} + C^\tau \chi_{ij} + C^\tau \tilde{\chi}_{ij} + C^\tau \Omega^X_{ij} + C^\tau \Omega^Y_{ij} + C^\tau \Omega^Z_{ij}$
      $C^T \text{Bell} = C^T \text{Bell} + C^T \text{Bell}_{ij}$
      $C^\tau \chi = C^\tau \chi + C^\tau \chi_{ij}$
      .
      .
      .
      $CQ^\tau[i] = CQ^\tau[i] + C^\tau[i, j]$
      $CQ^\tau[j] = CQ^\tau[j] + C^\tau[i, j]$
  for $m \in \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ **do**
    $C^T m_i = \binom{FI+Fm}{2}$
    $CQ^\tau[i] = CQ^\tau[i] + C^T m_i$
    $C^T m = C^T m + C^T m_i$
return C^τ, CM^τ, CQ^τ

Appendix C: Asymptotic scaling

In Fig. 4 a) we show the dependence of the number of groups on the total number of Pauli strings in the Hamiltonian of different molecules. We found that TPB obtains the largest number of groups for all molecules, while EM results in a significant reduction in the number of groups. HEEM obtains better results than TPB, but due to the constrain in the non-connected measurements, it does not reach the results of EM. However, in Fig. 4 b) we find a considerable improvement on the total number of CNOT gates. After transpiling, taking into account a real quantum device architecture, EM needs more CNOT gates to perform SWAP gates between nonconnected qubits. This problem is solved by using HEEM, which does not need any SWAP gate, resulting in less CNOT gates.

It is also important to consider the time needed to perform the grouping on a classical CPU, see Fig. 4 c). All the three algorithms begin with the construction of the Pauli graph, which contains information about the commutative Pauli strings. Once the graph is obtained, TPB uses LDFC for the graph coloring which time complexity is $\mathcal{O}(n^2)$, where n is the total number of Pauli strings. EM makes use of a more sophisticated algorithm, which needs to run through the graph and check if it can group a pair of terms with any of the existing base. This extra check results in a slower algorithm. Furthermore, HEEM also checks if the grouping is compatible with the chip's connectivity. However, despite the fact that the HEEM grouping is slower than the other methods, looking at the simulation time, Fig. 4 d), it is clear that the reduction in the simulation time is notorious. Even if TPB circuits do not require CNOT gates, the number of circuits to simulate is much larger than that of HEEM. On the other hand, EM grouping results in lower circuits, but the number of SWAP gates needed to perform the simulation grows much faster than that of HEEM, making it the slower algorithm to simulate.

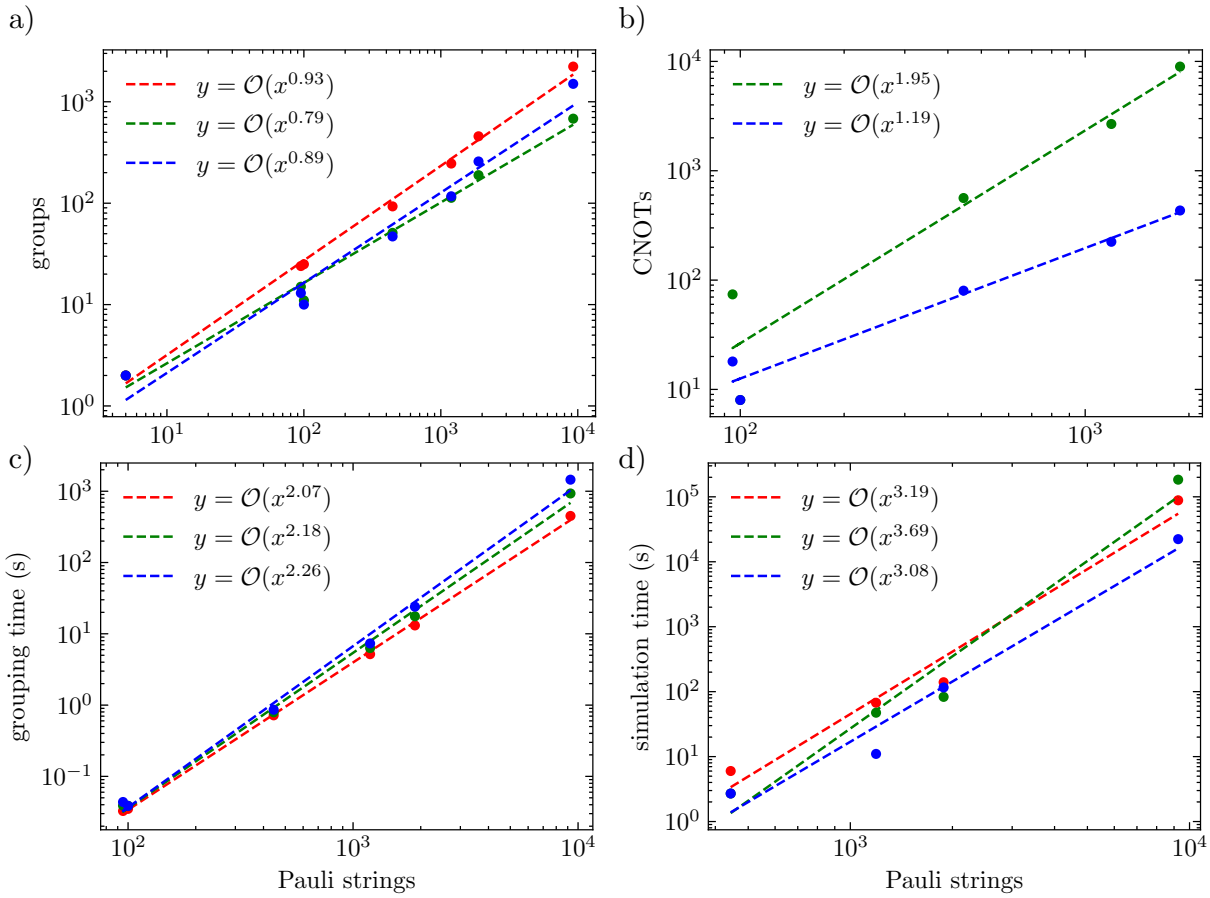


Figure 4. Dependence of a) the total number of groups, b) the number of CNOTs gates, c) CPU time for the grouping, and d) CPU time for the simulation of different molecules on the number of Pauli strings. The grouping algorithms used are TPB (red), EM (green), and HEEM (blue). The dashed lines represent a fit to the function $y = \beta x^{\alpha}$. Both in EM and in HEEM use *ibmq_montreal* connectivity. In d) the simulation mimics the architecture, base gate, and noise of the quantum device *ibmq_montreal*. Each simulation has a total of 2^{14} shots evenly distributed across all measurements in each grouping. The simulation is performed using the qasm HPC simulator provided by IBMQ. Transpiling and queue times are not included in these times.