

Leveraging randomized compiling for the quantum imaginary-time-evolution algorithm

Jean-Loup Ville^{1,*}, Alexis Morvan^{1,2,†}, Akel Hashim¹, Ravi K. Naik¹, Marie Lu¹, Bradley Mitchell¹, John-Mark Kreikebaum^{1,3}, Kevin P. O'Brien⁴, Joel J. Wallman^{5,6}, Ian Hincks⁶, Joseph Emerson^{5,6}, Ethan Smith², Ed Younis², Costin Iancu², David I. Santiago^{1,2} and Irfan Siddiqi^{1,2,3}

¹Quantum Nanoelectronics Laboratory, Dept. of Physics, University of California at Berkeley, Berkeley, California 94720, USA

²Computational Research Division, Lawrence Berkeley National Lab, Berkeley, California 94720, USA

³Materials Sciences Division, Lawrence Berkeley National Lab, Berkeley, California 94720, USA

⁴Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

⁵Inst. for Quantum Computing and Dept. of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

⁶Quantum Benchmark Inc., Kitchener, Ontario, Canada N2H 5G5



(Received 27 October 2021; revised 18 April 2022; accepted 31 May 2022; published 22 August 2022)

Recent progress in noisy intermediate-scale quantum (NISQ) hardware shows that quantum devices may be able to tackle complex problems even without error correction. However, coherent errors due to the increased complexity of these devices is an outstanding issue. They can accumulate through a circuit, making their impact on algorithms hard to predict and mitigate. Iterative algorithms like quantum imaginary time evolution are susceptible to these errors. This article presents the combination of both noise tailoring using randomized compiling and error mitigation with purification. We also show that cycle benchmarking gives an estimate of the reliability of the purification. We apply this method to the quantum imaginary time evolution of a transverse field Ising model and report an energy estimation error and a ground-state infidelity both below 1%. Our methodology is general and can be used for other algorithms and platforms. We show how combining noise tailoring and error mitigation will push forward the performance of NISQ devices.

DOI: [10.1103/PhysRevResearch.4.033140](https://doi.org/10.1103/PhysRevResearch.4.033140)

I. INTRODUCTION

To realize impactful application of noisy intermediate-scale quantum (NISQ) devices [1], error mitigation strategies have emerged as a principle focus of quantum information science. Unlike quantum error correction, which corrects errors as they occur, error mitigation uses postprocessing techniques to reduce the impact of errors on the results of an algorithm. These mitigation schemes are needed to tackle the noise and errors present in current quantum hardware. Many recently implemented algorithms have required some form of error mitigation with state-of-the-art hardware [2–6]. Several types of error mitigation can be distinguished: error extrapolation scales the rate of a specific known error in order to extrapolate the results to zero noise [3,5,7,8] at the expense of additional measurements and assumptions on the noise. Inverting error protocols characterize errors to then correct them with quasiprobabilities [8,9], requiring a precise and extensive characterization of the system using quantum process tomography [10] or gate set tomography [11]. Postselection protocols eliminate wrong output solutions by checking, for

example, an expected symmetry [12,13]. However additional measurement overhead, such as ancillary qubits, may be required for such postselection techniques. These additional quantum resources may limit scalability.

For quantum algorithms on NISQ hardware, one of the biggest challenges comes from coherent errors. Contrary to decoherence, the accumulation of coherent errors strongly depends on the circuit used. These errors arise due to the increasing complexity of quantum devices and originate from multiple mechanisms like crosstalk, frequency collision, drift, etc., making it difficult to track and compensate for their impact. On the contrary, decoherence processes are easier to predict and correct because these errors do not get amplified like coherent ones. To address the problem of coherent errors without significantly increasing the number of experimental repetitions per expectation value it is possible to tailor them into stochastic noise using Twirling properties, where two-qubit gates are sandwiched between twirling gates, and the circuit outcomes are statistically averaged over several twirled circuits. This technique usually keeps the number of repetitions constant but increases the number of circuits to be run. Twirling is a technique now widely known in the quantum information literature and is, for example, at the heart of randomized benchmarking [14–17]. Randomized compiling (RC) [18], which uses Pauli twirling, has been shown to improve the performance of quantum devices [19,20].

Common benchmarks of error mitigation techniques on NISQ devices are usually fixed-depth algorithms such as VQE [21–23] or QAOA [24–26]. Here we use the quantum imaginary-time-evolution (QITE) [27–30] algorithm to

*jlville@berkeley.edu

†morvan.alexis@gmail.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

benchmark the different error mitigation techniques. It is an iterative algorithm that approximates imaginary-time evolution with a unitary operation. In the limit of long imaginary time, the algorithm reaches the ground state of a given Hamiltonian. One advantage of QITE is that it generalizes easily to calculate finite temperature quantities [31,32]. Additionally it does not require *a priori* knowledge of an *Ansatz*, which can be difficult for variational algorithms [33–35]. Compared with VQE [36,37], each step of the QITE algorithm is more sensitive to noise, making it a good candidate to benchmark error mitigation protocols.

In this article, we use three out of eight transmon qubits on a fixed-frequency device with linear topology, as described in Refs. [20,38]. The single qubit gates are performed using the $ZXZXZ$ decomposition with virtual Z gates [39]. Our entanglement is based on the differential AC Stark shifts and realizes a CZ gate [40]. We combine noise tailoring with RC and mitigation with purification and show that using both improves the quality of the result beyond what one would expect from using each one separately. We attribute this performance improvement to the noise tailoring by RC that effectively maps the coherent errors into Pauli errors, which are simpler to handle and can be further approximated as a fully depolarizing error model. In Sec. II we discuss the implementation of RC with purification and give an estimation of how close the noise is to fully depolarizing using Cycle Benchmarking [41] and compare it to our hardware. In Sec. III, we then use this scheme to perform QITE on the transverse Ising field model with three qubits to benchmark the efficacy of this method. We conclude with a description of how to extend and complement our techniques with further mitigation schemes.

Recently, we became aware of comparable methods used in Ref. [42].

II. NOISE TAILORING WITH RANDOMIZED COMPILING

Twirling is a powerful technique that tailors the noise a circuit experiences when run on hardware. Its most notable use is the characterization of quantum processors with randomized benchmarking (RB) for qubits [14–17] and qudits [43,44]. Defining a twirl requires a twirling group \mathcal{G} that is usually the Pauli group or the Clifford group. For every cycle of the circuit—determined by the native gate-set—a gate from the twirl group is inserted. Twirling in the mathematical sense is achieved by averaging the circuit outcomes for all combinations of twirling gates. In practice, averaging over a few randomly sampled twirls is enough. When the twirling group is a single qubit group like the Pauli group \mathcal{P} , this step is done efficiently by compiling the twirling gate into the circuit’s single qubit gates, keeping the circuit’s depth unchanged in terms of multiqubit gates. For twirling to be useful for a specific circuit, one has to track all the twirl operations and invert it at the end of the circuit. Even for a simple twirling group like the Pauli group, this can be challenging if the circuit is made of non-Clifford gates. To circumvent this issue, protocols like randomized compiling (RC) have been proposed [18,45] and demonstrated [19,20]. In this protocol, the steps are separated between “easy” and “hard” cycles depending on their error rate. The hard cycles usually correspond to the cycles with multiqubit gates. The twirling gate is inverted through each

hard cycle instead of only at the final step. This approach avoids tracking all the twirling gates and the complex inversion gate at the end of the circuit, making it scalable to any number of qubits. However, it requires that any Pauli can be inverted through the entangling gates by another Pauli. It is the case for all the two-qubit gates which are locally equivalent to Cliffords, like the CNOT and the CZ, but not for entangling gates like the FSIM [46].

Pauli twirling and cycle benchmarking. Pauli twirling effectively changes the Pauli-transfer matrix (PTM) [47] of the error Λ . It can be shown that the error matrix Λ_P under Pauli twirling is simply the diagonal of the full PTM Λ (i.e., the off-diagonal terms of the error are suppressed). This emphasizes the advantage of RC over simply running a circuit: eliminating the off-diagonal terms of Λ makes any coherent interference of error impossible and thus increases the algorithms’ predictability. Several protocols are specifically designed to measure these Pauli errors [41,48–50]. In this article we use cycle benchmarking (CB) [41] as our main method for measuring the Pauli errors in our system. For each Pauli, CB measures the Pauli decay λ_P that corresponds to the diagonal of the Λ_P PTM. It can be used to exhaustively measure all Pauli channels—for small sized system—or to statistically sample from a large set of channels.

Even though the number of nonzero entries of Λ_P is reduced compared with that of Λ , there are still 4^N elements, making a complete characterization of all the terms not scalable as the number of qubits increases. However, CB demonstrates that sampling from these coefficients can give a good estimate of the behavior under randomized compiling. Introducing $\bar{\lambda}$ as their mean, the Pauli decays are also bounded by (see supplement of Ref. [41]):

$$2\bar{\lambda} - 1 \leq \lambda_P \leq 1 \quad \forall P \in \mathcal{P}. \quad (1)$$

In this article, we argue that the noise under RC with Pauli twirling can be approximated by a fully depolarizing noise model, where all the Pauli decays except the identity are equal, within a controlled approximation. In all experiments we have treated readout errors using standard readout mitigation by measuring the response matrix and using a least-square inversion to compensate for it in the processing of the data.

Experimental investigation of RC on random circuits. To demonstrate the properties of Pauli twirling under RC, we sample uniformly random two-qubit circuits [random in $SU(4)$] and compute the expectation values E_m of all the possible Pauli strings composed of Z and I . By compiling the rotations of the eigenbasis of an observable O into the last cycle of the circuit and by removing the identity part of this observable [i.e., requiring that $\text{Tr}(O) = 0$], we can always map the measurement of O to such a Pauli string. In Fig. 1 we plot the distribution of the mean errors on the expectation value measured E_m compared with its ideal value E as we increase the number of randomizations, thus coming closer to an ideal twirl. We have measured the expectation values of 75 different unitaries for each number of randomizations. To make the comparison fair, we have varied the number of randomly compiled circuits, keeping the total number of shots at 5000. The detailed experimental protocol is described in the Appendix. First, the spreading of the errors $|E_m - E|$ is reduced as can be seen in Fig. 1(a) as one would expect

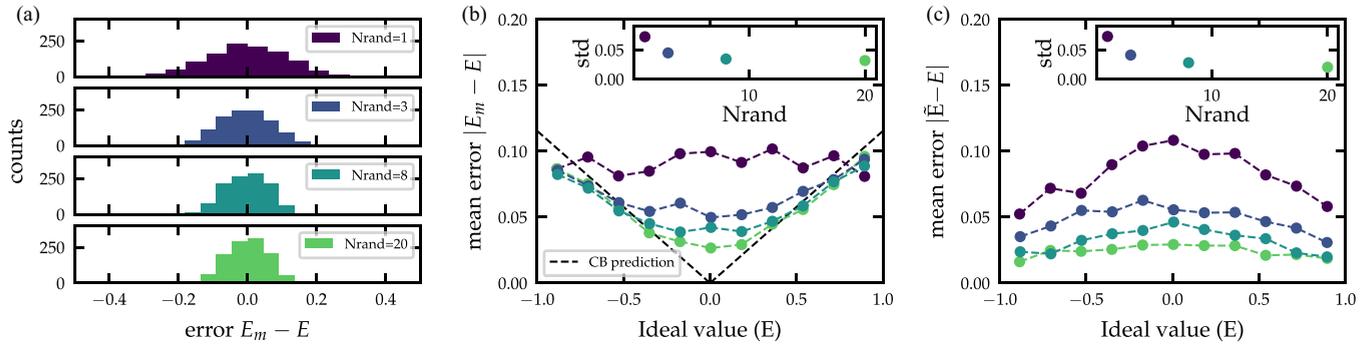


FIG. 1. Effect of randomized compiling on measured expectation values for random circuits using 6 CZs. Panel (a) shows the distribution of errors on the measured expectation values $|E_m - E|$ for the different numbers of randomizations. Panel (b) shows the same data but sorted by measured expectation values to show the increasing agreement with a depolarization model $E_m = \bar{\lambda}E$, with $\bar{\lambda}$ the mean of the Pauli decays measured by CB ($\bar{\lambda}_{CB} = 0.980$)—performed prior to the experiment. The standard deviation of the errors is similar for all points of same number of randomizations and is plotted in the inset. The data in panel (c) show the reduction of errors for all expectation values when using the purification formula (2). The standard deviation of the Pauli decays is $\text{std}(\lambda_p) = 0.003$, as shown at the top of Fig. 3.

from reducing the possibility of error accumulation. Second, in Fig. 1(b), the mean error without RC is flat as a function of the ideal value, while as we increase the number of randomizations, a linear dependency on the ideal value appears, corresponding to a fully depolarizing noise model.

Estimators for the average depolarization. Under a fully depolarizing noise model, the measured expectation value can be written as $E_m = \lambda E$. The scaling factor λ can also be understood as the length of the generalized Bloch vector. An estimator of λ can be constructed measuring the purity of the state at the end of a circuit. This, however, requires full tomography of the final state. An alternative estimator is the mean of the Pauli decays λ_p measured using CB. As it is a scalable protocol, it is possible to use it to estimate λ . In Fig. 1(b) we have plotted in dashed lines the average depolarization estimated from the mean of the Pauli decays measured with CB of the CZ gate, $\bar{\lambda}_{CB} = 0.980$, which shows good agreement. We can extract more information from the CB data: the distance between the two noise matrices Λ_d (depolarizing error) and Λ_p (twirled errors or Pauli errors) is exactly the standard deviation of the measured Pauli decays under CB. This provides very valuable *a priori* information on how well error mitigation protocols will work, based on the assumption that the error can be approximated by a depolarizing model.

Depolarizing errors can be mitigated by a simple purification. A fully depolarizing noise model is simple to mitigate. Knowing λ , the ideal expectation value can be recovered by rescaling all expectation values by this same coefficient λ . This technique has been used in nuclear magnetic resonance experiments and more recently in Ref. [51]. In Fig. 1(c), we have used a full tomography of the two-qubit state to extract the length of the generalized Bloch vector and then purified the expectations values measured with this information. In this case, the purification is given by

$$\tilde{E}_p = \lambda E_p \text{ with } \frac{1}{\lambda^2} = \frac{1}{2^N - 1} \sum_{P \in \mathcal{P}^{\otimes n} \setminus \{I^{\otimes n}\}} E_p^2. \quad (2)$$

Increasing the depth. In Fig. 2, we have increased the depth of two-qubit circuits. For two qubits there is only one hard

cycle, so the depth is the number of times this hard cycle is applied. The sampling is done by picking random unitaries from $SU(4)$ and using a KAK decomposition. For a depth-six circuit for example, we have randomly chosen two unitaries, decomposed them into our native gate set and then measured the concatenated circuits. We use this method to preserve the uniform sampling for all the different depths. The fidelity of a measured state ρ to a pure state σ can be expressed in terms of their Pauli expectation values:

$$F(\rho, \sigma) = \text{tr}(\rho\sigma) \quad (3)$$

$$= \frac{1}{2^N} \left(1 + \sum_{P \neq I^{\otimes N}} \rho_P \sigma_P \right) \quad (4)$$

$$= \frac{1}{2^N} + \left(1 - \frac{1}{2^N} \right) \lambda \cos \epsilon, \quad (5)$$

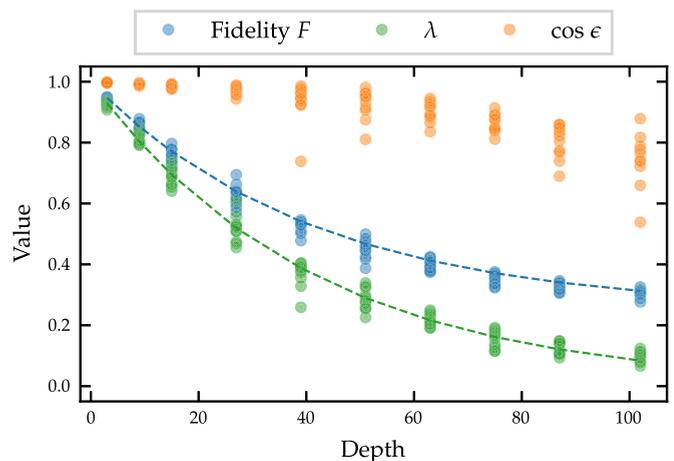


FIG. 2. Fidelity F of random states versus the number of CZ gates presented in blue points, together with the independent estimation from CB, using Eq. (7) as the dashed blue line. When these data were taken, the CB of this CZ gate was $\bar{\lambda}_{CB} = 0.976$. The length of the generalized Bloch vector λ , is shown in green, together with its CB estimation in dashed green. The remaining error separating the two contributions corresponds to an angle error as shown by Eq. (5).

where λ is the length of the Bloch vector given by Eq. (2) and ϵ is the angle between the two generalized Bloch vectors. Equation (5) indicates that two mechanisms can decrease the fidelity: a reduction of the Bloch vector length λ and a misalignment of the vectors axes giving a contribution $\cos \epsilon$. Purification or rescaling is intended to correct the first type of error but leaves untouched the second kind of error, or angle error. Assuming that the first kind of error is the limiting one, the fidelity can be approximated as

$$F(\rho, \sigma) \simeq \frac{1}{2^N} + \left(1 - \frac{1}{2^N}\right)\lambda. \quad (6)$$

For a circuit with the same hard cycle repeated n_{cycles} time and with an average Pauli decays $\bar{\lambda}$ we can simply write that

$$F(\rho, \sigma, n_{\text{cycles}}) \simeq \frac{1}{2^N} + \left(1 - \frac{1}{2^N}\right)\bar{\lambda}^{n_{\text{cycles}}}. \quad (7)$$

The measured fidelities obtained from full tomography for 10 different depths and 10 random circuits at each depth, randomly compiled 20 times are shown in Fig. 2. The estimation of Eq. (7) using $\bar{\lambda}$ from CB in dashed blue shows a very good agreement indicating that the length errors are indeed dominating. We also extract the length λ for each random circuit, and compare it to $\bar{\lambda}^{n_{\text{cycles}}}$. We also extract the residual error using the following equation:

$$\cos \epsilon = \frac{1}{2^N - 1} \sum_{P \neq I^{\otimes N}} \frac{\rho_P}{\lambda} \sigma_P. \quad (8)$$

This part of the error cannot be corrected with a simple purification. We note that the degradation of the fidelity due to the angle error is much slower than the part due to reduction of the Bloch vector length and that with purification, circuits with larger depth can be explored. This result emphasizes that randomized compilation makes circuit performance much more predictable, and that CB is a good tool for predicting circuit performance under RC.

Increasing the number of qubits. When increasing the number of qubits, several hard cycles need to be considered. In our case, for a linear topology of three qubits, we need to consider the three-qubit Pauli decays obtained by CB for the entangling gates between each qubit. Under randomized compiling the effective rescaling factor λ will be given by the product of the λ_i obtained for each hard cycle: $\lambda_{\text{eff}} = \lambda_1^{n_1} \lambda_2^{n_2}$ with n_i being the number of occurrences of the hard cycle. We note that universal circuits can be constructed using few different hard cycles and single qubit gates, reducing the number of hard cycles it is necessary to characterize. In Fig. 3, the spread of the Pauli decays λ_P are shown for the CZ gate used in Figs. 1 and 2, and for the CZ cycles of the three-qubits. We also emphasize that the number of Pauli decays needed to benchmark isolated two-qubit gates [15 for CZ(6, 5)] is much less than the number required to benchmark larger cycles containing idling spectator qubits [64 for CZ(5, 4) in parallel with I(6)]. We emphasize that exhaustive sampling of the Pauli decays is unnecessary because the mean and standard deviation can be estimated efficiently by randomly sampling the Pauli decays [41]. We notice that the bound from Eq. (1) is indeed valid for these data.

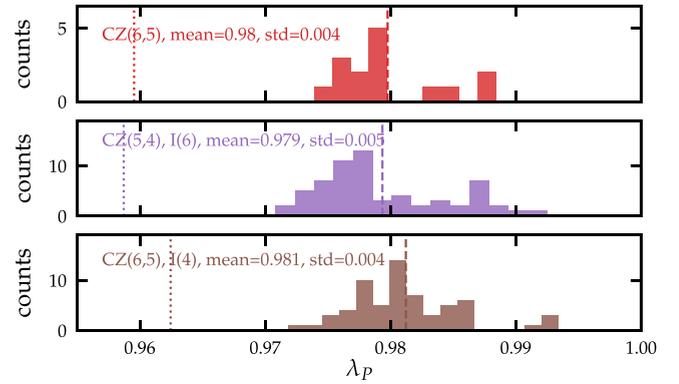


FIG. 3. Histograms of the Pauli decays λ_P for the different cycles. The first histogram shows the 15 different λ_P for the CZ gate of Fig. 1. The dashed line indicates the mean and the dotted line the lower bound of Eq. (1). The two other histograms show for comparison the corresponding parameters when characterizing the different three-qubit CZ hard cycles, including a remaining idling qubit. The spread of the 63 corresponding λ_P is shown for the two cycles used in the following QITE experiments.

III. APPLICATION TO THE QITE ALGORITHM

Imaginary time evolution is a classical iterative algorithm to find the ground state of a Hamiltonian. The key ingredient of this algorithm is that the imaginary time propagator $U(\beta) = \exp(-\beta\mathcal{H})$ —which is nonunitary—will converge to the ground state for large imaginary time, given that the initial state overlaps with the ground state [52,53]. In Ref. [27], the authors describe how to use a quantum computer to perform the imaginary-time evolution on NISQ hardware without ancilla qubits. The main idea is to normalize the evolution operator at every time step to make it a unitary evolution that can then be decomposed into gates. This can be done efficiently by simply solving a linear system, which is an easy task for classical computers, and thus QITE is free of the complex optimizations that arise in the VQE scheme [54]. The price to pay is that this algorithm is not a fixed depth circuit, but rather will increase the number of gates for every iteration. Recent experimental and theoretical works try to minimize this issue by aggressively reducing the number of steps needed to reach the ground state [28] or by compressing all the steps into a shorter circuit [55,56].

TFIM model. In this work, we concentrate on the transverse field Ising model (TFIM). This is a very well known model which has been investigated several times with the QITE and other algorithms [27,29,31,57] and is thus a proper benchmark for our error mitigation scheme. The TFIM Hamiltonian for a chain of N qubits is

$$\mathcal{H} = J \sum_{\langle ij \rangle} X_i X_j + h \sum_i Z_i, \quad (9)$$

where J is the interaction exchange between the nearest neighbors, h is the transverse field applied to the chain and $\langle ij \rangle$ indicates that the sum is over nearest neighbors. The state and the evolution operator at a given step can be written in the

Pauli Basis:

$$\rho = \sum_{P \in \mathcal{P}^{\otimes N}} \rho_P P \text{ and } U = \exp\left(-i \sum_{P \in \mathcal{P}^{\otimes N}} a_P P\right), \quad (10)$$

where ρ_P are the expectation values of the Paulis of the state ρ and a_P are the generators of the unitary U . We call support of the state the Paulis that have nonzero expectation values. The TFIM Hamiltonian presents several symmetries that allow us to reduce the problem from two perspectives: the construction of the unitary and the support of the ground state (described in the Appendix). In Motta *et al.* [27], a domain size D is introduced, which can be smaller than the full domain considered by the Hamiltonian, and the Trotterization happens over the different small domains D . This will be mandatory for bigger systems, but for the small systems considered here we will use D of the same size as the number of sites of the Hamiltonian.

Unitary and circuit construction. This section describes the protocol to synthesize each imaginary time step. The generators a_P are calculated via linear regression as discussed in Ref. [27]. The number of Pauli expectations values to measure and the number of generators depend on the size D of the domain considered in the QITE experiment. As our current processor has a limited number of qubits, we have chosen to consider a domain size equal to the total size of the system. This choice allows us to compress the circuit at every step and avoid stacking the gates. It enforces, however, a measurement of the state on its full support and makes the synthesis harder as it uses multiqubit gates with more than two qubits. Using the state-of-the-art circuit synthesis algorithm QSEARCH [58], we were able to run all of the three-qubit TFIM QITE steps with circuits containing always fewer than 12 CZs (see the Appendix for a comparison of the different synthesizers). We note that enforcing the symmetries on the unitary allowed us to drastically reduce the number of entangling gates. For larger qubit numbers the current approach will have to be improved, but we also expect that in the future, the synthesizers will continue to become more efficient. We also expect that the synthesis could be further tailored for the QITE algorithm. This type of synthesis has recently allowed us to find fixed depth *Ansätze* for some iterative algorithms and specific Hamiltonians [59].

Results. To showcase the error mitigation developed in the first section of this article, we have run the QITE algorithm on the three-qubit TFIM for several sets of parameters. In Fig. 4, we plot the QITE trajectory for the parameters $J = h = 1$ and calculate both the relative energy error and the infidelity of the measured ground state. For each QITE trajectory we varied the number of RC but fixed the total number of shots to show the effect of the different methods independently (three first symbols, in green, brown, and red). This shows that our method mitigates the jumps previously seen with this algorithm. We indicate on the plots the average of the last points as well as the standard deviations of the measurement after the ground state is reached. This is done in order to capture the stability of the algorithm when the ground state is found. On this plot we see that without randomized compiling—or error tailoring—nor error mitigation, the results are far away from the ground state. Using a purification technique improves the result significantly as it diminishes the impact

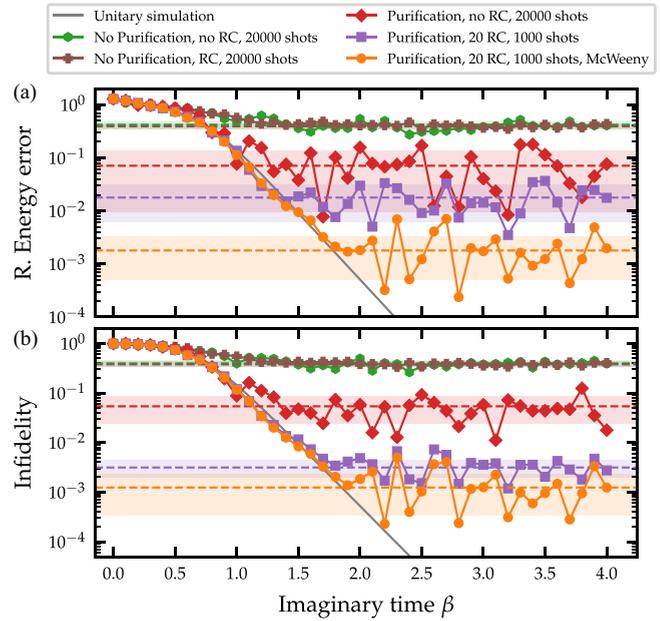


FIG. 4. QITE trajectories of relative energy error and infidelity to the ground state for the three-qubit Ising model, with $J = h = 1$. The number of shots and of randomized compilations per point is varied. The mean and standard deviation of the 10 last points are shown by a dashed line and a colored region. The different colors correspond to with or without RC, with the total number of shots kept constant at 20 000 to allow for a fair comparison, and with or without purification.

of incoherent errors introduced by noise tailoring via RC. We note here that iterative algorithms like QITE are particularly sensitive to errors on the expectation value as these values are necessary to determine the next circuit. We then have used both randomized compiling and the purification technique described in the previous section (purple squares). As we can see, the combination of both the noise tailoring and the error mitigation greatly improves the results. We further use McWeeny purification [60], also used for example in Ref. [4], corresponding to an iterative procedure projecting the measured state to the closest state of purity one (orange dots). As we increase the number of randomizations and the number of shots per randomization, we are able to push the precision to 0.2% for both the energy and the ground-state infidelity. It is also possible to compute the first-excited state, as described in the supplements.

Phase diagram. We then proceed to measure the phase diagram of the TFIM on three qubits as the external magnetic field h is swept. In Fig. 5 we plot the energies and the local magnetization as a function of the transverse field h for both the ground state and the first-excited state. For both these quantities, we have used the average over the last states as depicted in Fig. 4. We consistently get errors below 1% for both the energy and magnetization.

IV. CONCLUSIONS

In this paper, we have demonstrated that tailoring the errors with randomized compiling simplifies the noise process

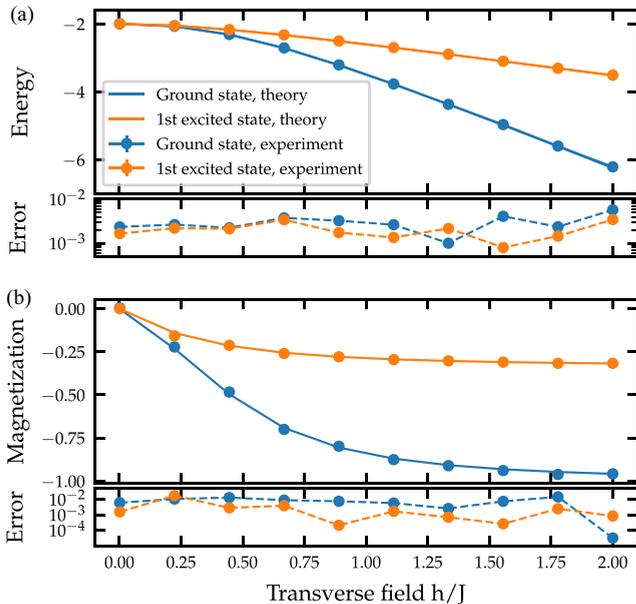


FIG. 5. Phase diagram for the three-qubit Ising model varying the h parameter. (a) Energy in units of E/J , for the ground state and first-excited state. The experiments use 10 randomized circuits and 1024 shots for each value of h . The reported energy corresponds to the mean of the five last points of the QITE evolution. Error bars corresponding to the standard deviation on these points are smaller than the markers. The error shown for each point corresponds to the relative error. (b) Measured magnetization of the same data, with the corresponding absolute error.

present on our quantum processor to a Pauli model. We also have shown that this error process is approximately a fully depolarizing noise. We have quantified the distance of our noise process from a fully depolarizing noise channel using the standard deviation of the Pauli decays obtained through cycle benchmarking. With this measure, we are able to predict whether simple purification techniques that compensate for this full depolarization can give an advantage. Since the spread of these Pauli decays is bounded, we believe that as the number of qubits increases, there will always be a regime where the noise under Pauli twirling can be approximated by a depolarizing model. In this work, we have concentrated on randomized compiling with Pauli twirling. The spread of Pauli decays depends on the choice of twirling group. Pauli twirling gives the maximum spread of Pauli decays with a different value for each, but is straightforward to implement. The Clifford group would lead to no spread at all but is impractical for more than two qubits. The Diehedral twirl [61] can be used with our native CZ gate and will reduce the spread as Pauli decays as those in X and Y will share the same values. This would reduce the distance to the depolarization channel and therefore potentially enhance the performance of purification techniques. We also foresee that using smart synthesizers that can exclude subspaces with large Pauli decays would help improve the performance of purification techniques.

We have demonstrated how an iterative algorithm—quantum imaginary time evolution—can benefit from the application of both a noise tailoring technique such as randomized compiling and an error mitigation technique such

as purification. The application of RC and purification results in an improvement over each technique used separately. The approximation made can be tested by measuring the spread of the Pauli decays using CB. In this article, we have concentrated on a simple purification scheme. However, it can be combined with sophisticated techniques, such as error extrapolation methods [3] or symmetric postselection [12,13] to even further enhance the accuracy of the results.

ACKNOWLEDGMENTS

This work was supported by the Quantum Test bed Program of the Advanced Scientific Computing Research for Basic Energy Sciences program, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and through the Office of Advanced Scientific Computing Research Accelerated Research for Quantum Computing Programs.

Jean-Loup Ville and Alexis Morvan contributed equally to this work.

I.H., J.J.W., and J.E. have a financial interest in Quantum Benchmark, Inc. and the use of True-Q software. The other authors declare no competing interests.

APPENDIX A: NOISE DESCRIPTION

Generic noise. A good way to understand how twirling works is considering the so-called Pauli transfer matrix (PTM) representation of quantum operators. The PTM is simply a matrix representation in the Pauli basis of the linear transformation on the density matrix. Hence, a PTM is a square matrix of size $4^N \times 4^N$, where N is the number of qubits. When considering a noisy implementation \tilde{A} of an ideal operation A , we can define the noisy part as $\Lambda = \tilde{A}A^{-1}$. For an ideal implementation, $\tilde{A} = A$ and Λ is the identity. In general, this error matrix can have terms on almost every position. Methods like process tomography [10] and gate set tomography [11] allow one to reconstruct this matrix using experimental measurements. The off-diagonal terms are specifically harmful for algorithms and prediction of performance as most of them will lead to accumulation of coherent errors. It can be visualized in the single-qubit case with a simple over-rotation: if the error is small for a single gate, repeating the same gate several times makes the error grow quadratically with the circuit depth. Additionally, the error is dependent on the circuit layout, making it hard to predict how well a given circuit will perform. This is one explanation for the gap between actual circuit performance results and what RB would predict [62].

APPENDIX B: VARIANCE OF THE EXPECTATION VALUE ERRORS

Measuring an expectation value. Most algorithms use quantum hardware to measure some specific expectation values that are relevant for the calculus of some energy or to determine the wave function. In this report we concentrate on the expectation value of Pauli operators. Following Ref. [41], we define our estimator as follows: For a given N -qubit Pauli operator Q , let \mathcal{B}_Q be the rotation that maps the computational basis to an eigenbasis of Q [e.g., a $R_y(\pi/2)$ for the single-qubit

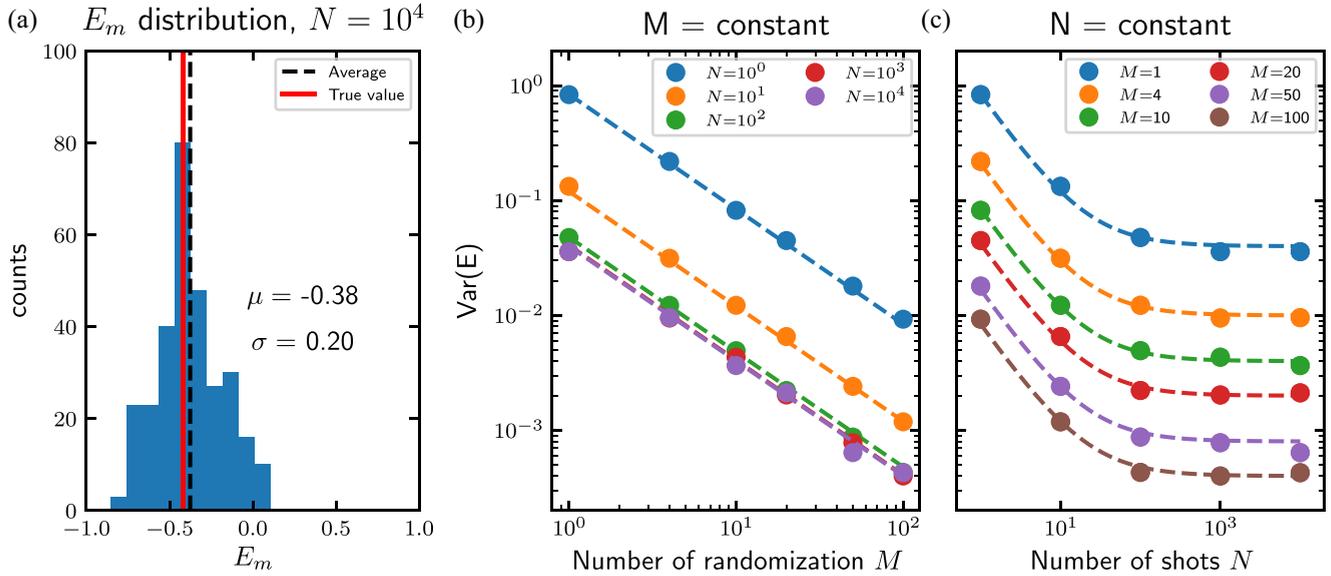


FIG. 6. Study of the distribution of the measured expectation values for randomly compiled circuits of the same unitary. (a) Dispersion of the single circuit estimator E_m in the limit of $N \gg 1$ so the single-shot noise is negligible. From this distribution, we extract the variance σ of the results from random implementation to another one. Panel (b) presents the evolution of the variance of the randomized compiled estimator E_{RC} as the number of randomizations increases. For each curve, the number of shots per random circuit is N . Panel (c) presents the same variance as a function of the number N of single shots per randomization. For large number of shots, the variances plateau to a value given by σ^2/M as expected from (B4). For both panels (b) and (c), the dashed lines are models following the Eq. (B4) and the extracted value from the distribution of panel (a).

operator X]. The measurement protocol gives an outcome z after the rotation in the eigenbasis with \mathcal{B}_Q^\dagger . The expectation value of Q can then be expressed as

$$\text{Tr}[Q\rho] = \sum_{z \in \mathbb{Z}_2^N} \text{Tr}[\mathcal{B}_Q(|z\rangle\langle z|)Q] \text{Pr}(z|Q), \quad (\text{B1})$$

which amounts to a weighted sum over the population measurement. Due to the discrete nature of the measurement (each single-shot yields a bit string), one needs to repeat several times the same circuit (or sequence) and average in order to have an estimate of the expectation value.

Theoretical estimation of the variance of the errors. For a given sequence, the variance evolves as

$$\text{Var}[E] = \frac{(1 - E)(1 + E)}{N_{\text{shot}}}, \quad (\text{B2})$$

where N_{shot} is the number of repetitions of the measurement and E is the expectation value. This variance is maximal for $E = 0$ and minimal for $|E| = 1$. Usually, we experimentally go to large enough N_{shot} to eliminate this source of noise.

Even though the variance is reduced by increasing the total number of shots, it is not clear that the measured expectation value is the right one. Actually, due to noisy gates, this estimator is biased. This bias depends strongly on the given implementation of the circuit. One way to construct a less biased estimator is to randomly sample from a set of circuits that realizes the same unitary, i.e., the total operation is identical but the circuits differ. Randomized compiling [18]—or Pauli twirling—gives a framework to do this. The implementation of the same operation only differs by the insertion of Pauli gates. The estimator for a given expectation value is then given

by the average over all the implementations:

$$E_{RC} = \frac{1}{M} \sum_{m=1}^M E_m, \quad (\text{B3})$$

where each E_m correspond to a different circuit. The advantage of using this approach is that the bias of the E_{RC} estimator is much more predictable, allowing one to compensate for it. Under this assumption, the variance of the E_{RC} estimator is given by [63]

$$\text{Var}[E_{RC}] = \frac{1}{M} \left(\frac{(1 - E_{RC})(1 + E_{RC})}{N} + \frac{N - 1}{N} \sigma^2 \right). \quad (\text{B4})$$

As pointed out in Ref. [63], the main information from this equation is that the total variance goes to zero as the number of circuits increases, but asymptotes to the finite value σ^2/M if the number of randomization is fixed and we increase the number of single-shot N . We have probed this with an experimental dataset in Fig. 6. When choosing between more random circuits or more single shots, the best choice is to use more random circuits, with the best situation being one shot per random circuit ($N = 1$). However, from an experimental perspective, it is not time efficient to run one shot per random circuit as the upload time to the control hardware is time consuming whereas repeating the same random sequence is fast. From this experiment, we can see that using up to a few hundred of single shots per random circuits gives a good compromise. We can also see that getting more than 10^3 shots does not improve the variance of the result for a fixed number of randomizations. This is of practical use when designing the RC sequence for a given circuit.

TABLE I. Comparison of the maximum number of entangling gates obtained with different synthesizers for N qubits. QSEARCH [58] and QFAST [64] and the generic tool Isometry on Qiskit [65] for a QITE simulation of a TFIM of N sites with the parameters $J = h = 1$.

N	QISKIT	QFAST	QSEARCH
2	3	3	3
3	30–35	10–12	7–12
4	160–200	70–80	30–50

APPENDIX C: DETAILED EXPERIMENTAL PROTOCOL FOR FIG. 1

In practical settings, one is interested in measuring expectation values. To study the effect of randomized compiling when measuring expectation values we have run a series of state tomographies performed as follows:

- (1) Draw a random unitary from a uniform sampling of $SU(4)$.
- (2) Decompose the unitary into a circuit using the KAK decomposition (3 CZ gates). Repeat until reaching the wanted depth (6 CZ gates for Fig. 1 of the main text).
- (3) Create nine circuits by appending the measurement rotation at the end of the circuit.
- (4) For each circuit generate M randomly compiled circuits.
- (5) Run each of the $9 \times M$ circuits N time to gather statistics.
- (6) Calculate the expectation value for each Pauli for each circuit (15 per unitary).

APPENDIX D: DIFFERENT SYNTHESIZERS FOR CIRCUIT CONSTRUCTION

As explained in the *Unitary and circuit construction* section of the main text, the gates which should have been added from the QITE algorithm are concatenated into the circuit to avoid increasing its depth. This is done by using a circuit generator from the computed unitary. We report in Table I the number of entangling gates for different synthesizers. QSEARCH allows us to set the numerical accuracy of the synthesis of the unitary, and it is possible to improve the computational time by reducing the accuracy while still not limiting the ground-state fidelity measured on the hardware.

APPENDIX E: SYMMETRIES OF THE HAMILTONIANS

We describe here how several symmetries of the TFIM Hamiltonian have been used to reduce the number measurements needed and of generators. This can apply to many Hamiltonians.

\mathbb{Z}_2 symmetry. Let us first consider the so-called \mathbb{Z}_2 symmetry: The Hamiltonian from Eq. (9) commutes with the operator $Z^{\otimes n}$. This symmetry divides the full Hilbert space into two eigenspaces with, for any state $|S\rangle$, $Z^{\otimes n}|S\rangle = +|S\rangle$ or $Z^{\otimes n}|S\rangle = -|S\rangle$. All the eigenstates of \mathcal{H} have to be in either of these subspaces. This property will both restrain the support of the possible ground states and force the QITE evolution operators to preserve the parity with respect to this symmetry. To reduce the support of the ground state, we recall that two Pauli operators can either commute or anticommute. For the Pauli operators that anticommute with the symmetry S , the expectation value of this Pauli on an eigenstate of the Hamiltonian is necessarily zero: $\langle GS|P|GS\rangle = \text{Tr}(\rho P) = \rho_P = 0$. If we require all the steps to fall within the symmetric subspace, this will also enforce that U commutes with the symmetry S . Developing the evolution to the first order, we find that U commutes with S if and only if $a_P = 0 \forall P \in \mathcal{P}$ such that $\{P, S\} = 0$. This simplifies the synthesis as the number of generators to consider is reduced by a factor two. For the support of the ground state, if we know the sign of the parity s , usually found using a classical algorithm, we can further constraint the support by noting that $\langle SP\rangle = s\langle P\rangle$. The number of free parameters for the ground state is thus reduced by a factor four compared with the full Hilbert space size.

Time-reversal symmetry. As the TFIM Hamiltonian \mathcal{H} is real, it is invariant by time-reversal symmetry. This adds another symmetry to consider. The corresponding symmetry operator is $T = K$, with K the complex conjugation. The unitary evolution at each step has to commute with K , which implies that the generator has to anticommute. This means that the support of the generators is included in the Pauli matrices that anticommute with K : $a_P = 0 \forall P \in \mathcal{P}$ such that $[P, K] = 0$. This implies that the set of allowed generators are the Pauli strings with an odd number of Y . It also means that the ground state should be an eigenvector of K meaning that the support of the eigenstate has to commute with K . In other words, the eigenstate support set is intersected by the Pauli strings that have a even number of Y . We note that for this specific symmetry, the set of generators and the set of support for the ground state are disjoint.

APPENDIX F: FIRST EXCITED STATE AND HIGHER-ENERGY LEVELS WITH THE QITE ALGORITHM

The QITE algorithm offers an efficient way to calculate the higher-energy states. When the ground state $|GS\rangle$ is determined without considering the symmetries, one idea is to add to the Hamiltonian \mathcal{H} a term proportional to the ground state in order to make the first-excited state into a ground state: $\mathcal{H} \rightarrow \mathcal{H} + \alpha|GS\rangle\langle GS|$ with a coefficient α large enough. It is also possible for the first-excited state to make use of the symmetry: the ground and first-excited states should have opposite parities. Then using the same QITE algorithm, but changing the initial state parity allows us to find the first excited state. This is what was done for Fig. 5 of the main text.

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

[2] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational

- quantum eigensolver for small molecules and quantum magnets, *Nature (London)* **549**, 242 (2017).
- [3] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Error mitigation extends the computational reach of a noisy quantum processor, *Nature (London)* **567**, 491 (2019).
- [4] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley *et al.*, Hartree-Fock on a superconducting qubit quantum computer, *Science* **369**, 1084 (2020).
- [5] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, Cloud Quantum Computing of an Atomic Nucleus, *Phys. Rev. Lett.* **120**, 210501 (2018).
- [6] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, *J. Phys. Soc. Jpn.* **90**, 032001 (2021).
- [7] Y. Li and S. C. Benjamin, Efficient Variational Quantum Simulator Incorporating Active Error Minimization, *Phys. Rev. X* **7**, 021050 (2017).
- [8] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, *Phys. Rev. Lett.* **119**, 180509 (2017).
- [9] S. Zhang, Y. Lu, K. Zhang, W. Chen, Y. Li, J.-N. Zhang, and K. Kim, Error-mitigated quantum gates exceeding physical fidelities in a trapped-ion system, *Nat. Commun.* **11**, 587 (2020).
- [10] I. L. Chuang and M. A. Nielsen, Prescription for experimental determination of the dynamics of a quantum black box, *J. Mod. Opt.* **44**, 2455 (1997).
- [11] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography, *Nat. Commun.* **8**, 14485 (2017).
- [12] S. McArdle, X. Yuan, and S. Benjamin, Error-Mitigated Digital Quantum Simulation, *Phys. Rev. Lett.* **122**, 180501 (2019).
- [13] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, Low-cost error mitigation by symmetry verification, *Phys. Rev. A* **98**, 062339 (2018).
- [14] J. Emerson, R. Alicki, and K. Życzkowski, Scalable noise estimation with random unitary operators, *J. Opt. B: Quantum Semiclassical Opt.* **7**, S347 (2005).
- [15] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, Randomized benchmarking of quantum gates, *Phys. Rev. A* **77**, 012307 (2008).
- [16] E. Magesan, J. M. Gambetta, and J. Emerson, Scalable and Robust Randomized Benchmarking of Quantum Processes, *Phys. Rev. Lett.* **106**, 180504 (2011).
- [17] E. Magesan, J. M. Gambetta, and J. Emerson, Characterizing quantum gates via randomized benchmarking, *Phys. Rev. A* **85**, 042311 (2012).
- [18] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, *Phys. Rev. A* **94**, 052325 (2016).
- [19] M. Ware, G. Ribeill, D. Risté, C. A. Ryan, B. Johnson, and M. P. da Silva, Experimental Pauli-frame randomization on a superconducting qubit, *Phys. Rev. A* **103**, 042604 (2021).
- [20] A. Hashim, R. K. Naik, A. Morvan, J.-L. Ville, B. Mitchell, J. M. Kreikebaum, M. Davis, E. Smith, C. Iancu, K. P. O'Brien, I. Hincks, J. J. Wallman, J. Emerson, and I. Siddiqi, Randomized Compiling for Scalable Quantum Computing on a Noisy Superconducting Quantum Processor, *Phys. Rev. X* **11**, 041039 (2021).
- [21] 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**, 4213 (2014).
- [22] A. J. McCaskey, Z. P. Parks, J. Jakowski, S. V. Moore, T. D. Morris, T. S. Humble, and R. C. Pooser, Quantum chemistry as a benchmark for near-term quantum computers, *npj Quantum Inf.* **5**, 99 (2019).
- [23] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, Computation of Molecular Spectra on a Quantum Processor with an Error-Resilient Algorithm, *Phys. Rev. X* **8**, 011021 (2018).
- [24] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, *arXiv:1411.4028*.
- [25] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, *Nat. Phys.* **17**, 332 (2021).
- [26] N. Lacroix, C. Hellings, C. K. Andersen, A. Di Paolo, A. Remm, S. Lazar, S. Krinner, G. J. Norris, M. Gabureac, J. Heinsoo, A. Blais, C. Eichler, and A. Wallraff, Improving the performance of deep quantum optimization algorithms with continuous gate sets, *PRX Quantum* **1**, 110304 (2020).
- [27] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. K.-L. Chan, Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution, *Nat. Phys.* **16**, 205 (2020).
- [28] K. Yeter-Aydeniz, R. C. Pooser, and G. Siopsis, Practical quantum computation of chemical and nuclear energy levels using quantum imaginary time evolution and Lanczos algorithms, *npj Quantum Inf.* **6**, 63 (2020).
- [29] K. Y. Aydeniz, G. Siopsis, and R. C. Pooser, Scattering in the Ising model with the quantum Lanczos algorithm, *New J. Phys.* **23**, 043033 (2021).
- [30] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational ansatz-based quantum simulation of imaginary time evolution, *npj Quantum Inf.* **5**, 75 (2019).
- [31] S.-N. Sun, M. Motta, R. N. Tazhigulov, A. T. K. Tan, G. K.-L. Chan, and A. J. Minnich, Quantum computation of finite-temperature static and dynamical properties of spin systems using quantum imaginary time evolution, *PRX Quantum* **2**, 010317 (2021).
- [32] L. Bassman, K. Klymko, N. M. Tubman, and W. A. de Jong, Computing free energies with fluctuation relations on quantum computers, *arXiv:2103.09846*.
- [33] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [34] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, *Nat. Commun.* **10**, 3007 (2019).
- [35] A. Choquette, A. Di Paolo, P. K. Barkoutsos, D. Sénéchal, I. Tavernelli, and A. Blais, Quantum-optimal-control-inspired

- ansatz for variational quantum algorithms, *Phys. Rev. Research* **3**, 023092 (2021).
- [36] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus *et al.*, Scalable Quantum Simulation of Molecular Energies, *Phys. Rev. X* **6**, 031007 (2016).
- [37] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states, *Phys. Rev. A* **95**, 042308 (2017).
- [38] M. Blok, V. Ramasesh, T. Schuster, K. O'Brien, J. Kreikebaum, D. Dahlen, A. Morvan, B. Yoshida, N. Yao, and I. Siddiqi, Quantum Information Scrambling on a Superconducting Qutrit Processor, *Phys. Rev. X* **11**, 021010 (2021).
- [39] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient Z gates for quantum computing, *Phys. Rev. A* **96**, 022330 (2017).
- [40] B. K. Mitchell, R. K. Naik, A. Morvan, A. Hashim, J. M. Kreikebaum, B. Marinelli, W. Lavrijsen, K. Nowrouzi, D. I. Santiago, and I. Siddiqi, Hardware-Efficient Microwave-Activated Tunable Coupling between Superconducting Qubits, *Phys. Rev. Lett.* **127**, 200502 (2021).
- [41] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, Characterizing large-scale quantum computers via cycle benchmarking, *Nat. Commun.* **10**, 5347 (2019).
- [42] M. Urbanek, B. Nachman, V. R. Pascuzzi, A. He, C. W. Bauer, and W. A. de Jong, Mitigating depolarizing noise on quantum computers with noise-estimation circuits, *Phys. Rev. Lett.* **127**, 270502 (2021).
- [43] A. Morvan, V. V. Ramasesh, M. S. Blok, J. M. Kreikebaum, K. O'Brien, L. Chen, B. K. Mitchell, R. K. Naik, D. I. Santiago, and I. Siddiqi, Qutrit Randomized Benchmarking, *Phys. Rev. Lett.* **126**, 210504 (2021).
- [44] M. Jafarzadeh, Y.-D. Wu, Y. R. Sanders, and B. C. Sanders, Randomized benchmarking for qudit Clifford gates, *New J. Phys.* **22**, 063014 (2020).
- [45] S. Ferracin, T. Kapourniotis, and A. Datta, Accrediting outputs of noisy intermediate-scale quantum computing devices, *New J. Phys.* **21**, 113038 (2019).
- [46] B. Foxen, C. Neill, A. Dunsworth, P. Roushan, B. Chiaro, A. Megrant, J. Kelly, Z. Chen, K. Satzinger, R. Barends *et al.*, Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms, *Phys. Rev. Lett.* **125**, 120504 (2020).
- [47] J. M. Chow, J. M. Gambetta, A. D. Córcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. R. Rozen *et al.*, Universal Quantum Gate Set Approaching Fault-Tolerant Thresholds with Superconducting Qubits, *Phys. Rev. Lett.* **109**, 060501 (2012).
- [48] R. Harper, S. T. Flammia, and J. J. Wallman, Efficient learning of quantum noise, *Nat. Phys.* **16**, 1184 (2020).
- [49] R. Harper, W. Yu, and S. T. Flammia, Fast estimation of sparse quantum noise, *PRX Quantum* **2**, 010322 (2021).
- [50] S. T. Flammia and J. J. Wallman, Efficient estimation of Pauli channels, *ACM Trans. Quantum Comput.* **1**, 1 (2020).
- [51] J. Vovrosh, K. E. Khosla, S. Greenaway, C. Self, M. S. Kim, and J. Knolle, Simple mitigation of global depolarizing errors in quantum simulations, *Phys. Rev. E* **104**, 035309 (2021).
- [52] S. K. Adhikari, Numerical solution of the two-dimensional Gross-Pitaevskii equation for trapped interacting atoms, *Phys. Lett. A* **265**, 91 (2000).
- [53] J. Auer, E. Krotscheck, and S. A. Chin, A fourth-order real-space algorithm for solving local Schrödinger equations, *J. Chem. Phys.* **115**, 6841 (2001).
- [54] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong, Classical optimizers for noisy intermediate-scale quantum devices, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Piscataway, NJ, 2020), pp. 267–277.
- [55] S.-H. Lin, R. Dilip, A. G. Green, A. Smith, and F. Pollmann, Real- and imaginary-time evolution with compressed quantum circuits, *PRX Quantum* **2**, 010342 (2021).
- [56] N. Gomes, F. Zhang, N. F. Berthussen, C.-Z. Wang, K.-M. Ho, P. P. Orth, and Y. Yao, Efficient step-merged quantum imaginary time evolution algorithm for quantum chemistry, *J. Chem. Theory Comput.* **16**, 6256 (2020).
- [57] A. Cervera-Lierta, Exact Ising model simulation on a quantum computer, *Quantum* **2**, 114 (2018).
- [58] M. G. Davis, E. Smith, A. Tudor, K. Sen, I. Siddiqi, and C. Iancu, Towards optimal topology aware quantum circuit synthesis, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Piscataway, NJ, 2020), pp. 223–234.
- [59] L. Bassman, R. V. Beeumen, E. Younis, E. Smith, C. Iancu, and W. A. de Jong, Constant-depth circuits for dynamic simulations of materials on quantum computers, *arXiv:2103.07429*.
- [60] R. McWeeny, Some recent advances in density matrix theory, *Rev. Mod. Phys.* **32**, 335 (1960).
- [61] A. Carignan-Dugas, J. J. Wallman, and J. Emerson, Characterizing universal gate sets via dihedral benchmarking, *Phys. Rev. A* **92**, 060302(R) (2015).
- [62] T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout, Measuring the capabilities of quantum computers, *Nat. Phys.* **18**, 75 (2021).
- [63] I. Hincks, J. J. Wallman, C. Ferrie, C. Granade, and D. G. Cory, Bayesian inference for randomized benchmarking protocols, *arXiv:1802.00401*.
- [64] E. Younis, K. Sen, K. Yelick, and C. Iancu, QFAST: Quantum synthesis using a Hierarchical continuous circuit space, *arXiv:2003.04462*.
- [65] G. Aleksandrowicz *et al.*, Qiskit: An open-source framework for quantum computing (2019).