



# Improved quantum computation using operator backpropagation



Bryce Fuller<sup>1</sup> ✉, Minh C. Tran<sup>1</sup>, Danylo Lykov<sup>2,3</sup>, Caleb Johnson<sup>1</sup>, Max Rossmannek<sup>4</sup>, Ken Xuan Wei<sup>1</sup>, Andre He<sup>1</sup>, Youngseok Kim<sup>1</sup>, DinhDuy Vu<sup>5</sup>, Kunal Sharma<sup>1</sup>, Yuri Alexeev<sup>2,3</sup>, Abhinav Kandala<sup>1</sup> & Antonio Mezzacapo<sup>1</sup>

Decoherence of quantum hardware is currently limiting its practical applications. At the same time, classical algorithms for simulating quantum circuits have progressed substantially. Here, we demonstrate a hybrid framework that integrates classical simulations with quantum hardware to improve the computation of an observable's expectation value by reducing the quantum circuit depth. In this framework, a quantum circuit is partitioned into two subcircuits: one that describes the backpropagated Heisenberg evolution of an observable, executed on a classical computer, while the other is a Schrödinger evolution run on quantum processors. The overall effect is to reduce the depths of the circuits executed on quantum devices and enable the recovery of expectation values at intermediate times throughout the classically backpropagated circuit, trading this with classical overhead and an increased number of circuit executions. We demonstrate the effectiveness of this method on a Hamiltonian simulation problem, achieving more accurate expectation value estimates compared to using quantum hardware alone.

Quantum algorithms promise significant advantages over classical methods in many applications such as Hamiltonian simulation<sup>1</sup> and solving systems of linear equations<sup>2</sup>. These advantages can often only be realized for sufficiently large problem instances and typically require coherent implementations of deep quantum circuits. However, decoherence of current quantum hardware limits their application to short-depth quantum circuits. Error mitigation<sup>3–5</sup> has been used to enable accurate calculations on current quantum hardware at a scale beyond brute force classical computation<sup>6</sup>. These methods typically incur a sampling overhead that is exponential in the depth of the circuit.

In response to the cost of error mitigation, recent experimental demonstrations aim to optimize the performance and resource overhead of quantum experiments by leveraging a variety of classical techniques to reduce the depth of executed quantum circuits. For example, advancements in transpilation have led to more efficient swap routing<sup>7,8</sup>, which can result in shallower circuits to execute. Tensor networks have been used in conjunction with classical optimizers to improve the accuracy of expectation values for time evolution problems<sup>9</sup>. Hybrid multi-product formulas<sup>10–12</sup> enable performing Hamiltonian time evolution using an ensemble of shallower quantum circuits. Recent algorithms for approximate quantum compiling utilize tensor networks and a classical optimizer to compress deep Trotterized time-evolution circuits into shallower approximations<sup>12–17</sup>.

Additionally, recent works have studied the application of feed-forward measurements to circuit compilation in the context of circuit cutting<sup>18</sup> and measurement-based state preparation<sup>19,20</sup>.

These rapid developments highlight an ongoing need to discover new algorithms to reduce the depth of quantum experiments.

In this manuscript, we introduce a framework to reduce the depth of quantum circuits using classical simulation algorithms based on Clifford perturbation theory (CPT)<sup>21</sup>. We then apply this framework using the Qiskit Addon for operator backpropagation<sup>22</sup> and execute experiments on quantum hardware to observe the reduction in error which can be achieved for a utility scale Hamiltonian time dynamics experiment. CPT-based algorithms classically compute the expectation value of an observable by *back-propagating* it, i.e. evolving the observable in the Heisenberg picture through the gates of the circuit in reverse order, starting from the last gate of the circuit. Provided that the circuit consists of a small number of non-Clifford operations, the Pauli decompositions of the backpropagated observable can be computed exactly by a classical computer in reasonable runtime. For circuits with a large proportion of non-Clifford gates the cost of exact backpropagation grows intractably with circuit depth; however, by allowing for some approximation error a wider range of circuits can be backpropagated in practice. Such a strategy has been successful in approximating the outcomes of recent utility-scale experiments<sup>23</sup>. Theoretically, algorithms

<sup>1</sup>IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, NY, USA. <sup>2</sup>Argonne National Laboratory, Lemont, IL, USA. <sup>3</sup>NVIDIA Corp, Santa Clara, CA, USA. <sup>4</sup>IBM Quantum, IBM Research – Zurich, Rüschlikon, Switzerland. <sup>5</sup>Department of Physics, Harvard University, Cambridge, MA, USA.

✉ e-mail: [Bryce.Fuller@ibm.com](mailto:Bryce.Fuller@ibm.com)

based on CPT have been shown to be asymptotically efficient for several interesting classes of quantum circuits, including those that are noisy with a random input<sup>24</sup> or those that consist of random single-qubit operations<sup>25</sup>.

In contrast to these asymptotic analyses, this work uses CPT to approximate the Heisenberg evolution for explicit circuit instances while tracking the accuracy of the calculation using a combination of typical-case error bounds<sup>24</sup> and the triangle inequality.

In our framework, a quantum circuit is split into two subcircuits  $U_Q$  and  $U_C$  (Fig. 1). The observable of interest is backpropagated under  $U_C$  and decomposed as a linear combination of Pauli operators. These Pauli operators are then measured on the quantum state prepared by  $U_Q$ . In general, the number of Pauli operators grows with the depth of the backpropagated subcircuit. Thus, *operator backpropagation* (OBP) allows one to reduce circuit depth in exchange for a classical overhead and an increase in the total number of circuits executed on quantum hardware. Because OBP calculations can become classically expensive and lend themselves to distributed implementation, our method is amenable to be run in quantum-centric supercomputing environments<sup>26</sup>. In Section II, we introduce the OBP framework and demonstrate an application of OBP for improving quantum simulation. In particular, we show that OBP helps reduce the error in computing the expectation values of observables for circuits with up to 127 qubits and 4896 two qubit gates. Hence, given a fixed error tolerance, OBP enables the computation of expectation values for deeper quantum circuits than a purely-quantum approach. Lastly, in Section III we describe the OBP framework, including details of the CPT algorithm, the truncation strategies employed, and strategies for parallelizing the CPT calculation across a multi-node architecture.

## Results

### Operator backpropagation

Many quantum algorithms rely on measuring operator expectation values with respect to states prepared on quantum devices. Specifically, we consider problems of estimating

$$\langle O \rangle_{U|\psi} \equiv \langle \psi | U^\dagger O U | \psi \rangle, \tag{1}$$

given a quantum state  $|\psi\rangle$ , a quantum circuit  $U$ , and an observable  $O$ . Without loss of generality, we assume that  $O$  is a traceless multi-qubit Pauli operator. In near-term experiments, the depths of quantum circuits for which the expectation values can be faithfully recovered are constrained by the error of the quantum devices, limiting the size of experiments that can be performed.

Parallel to the advancements in quantum hardware, various classical algorithms have been developed for numerically computing the expectation value in Eq. (1). For problems considered in recent experiments, these classical algorithms often perform better or as well as algorithms executed on state-of-the-art quantum devices<sup>21,27</sup>.

Despite this, the classical complexity of estimating arbitrary expectation values grows exponentially with problem size and, thus, will become out of reach for general classical algorithms even on the largest supercomputers.

To distribute the expectation value problem in Eq. (1) between a quantum device and a classical simulator, we consider a decomposition of  $U = U_C U_Q$  into two subcircuits  $U_C$  and  $U_Q$ . The classical simulator first computes  $O' \equiv U_C^\dagger O U_C$ —the version of  $O$  evolved backward through the circuit  $U_C^\dagger$ . One then prepares the initial state  $|\psi\rangle$  on the quantum hardware, applies the circuit  $U_Q$ , and measures the expectation value of  $O'$ .

It is straightforward to verify that the result  $\langle \psi | U_Q^\dagger O' U_Q | \psi \rangle = \langle \psi | U^\dagger O U | \psi \rangle$  is the desired expectation value in Eq. (1).

Standard quantum hardware can measure the expectation values of observables that are diagonal in a local measurement basis. To measure the expectation value of  $O'$ , we require that the classical simulator decomposes it

in the Pauli basis, i.e.

$$O' = \sum_P c_P P, \tag{2}$$

where  $P \in \{I, X, Y, Z\}^{\otimes n}$  are multi-qubit Pauli operators on  $n$  qubits and the real numbers  $c_P = \text{Tr}(O'P)/2^n$  are the coefficients of the decomposition. We call the process of approximately evolving  $O$  through  $U_C^\dagger$  *operator backpropagation* (OBP). Given the backpropagated observable  $O'$ , we measure the expectation value of each Pauli in the decomposition and reconstruct the expectation value of  $O'$  by

$$\langle \psi | U_Q^\dagger O' U_Q | \psi \rangle = \sum_P c_P \langle \psi | U_Q^\dagger P U_Q | \psi \rangle. \tag{3}$$

We summarize these steps in Fig. 1.

Our framework offers a trade-off between the required circuit depth and the number of circuit executions which are needed to compute  $\langle O \rangle_{U|\psi}$ . Because the circuit executed on quantum hardware  $U_Q$  will be shallower than the original circuit, the resources needed to error mitigate each individual circuit are lowered. In exchange, the number of distinct circuits which must be executed increases with the number of Pauli operators that comprise  $O'$  in Eq. (2). In general, the number of Pauli measurements and the error-mitigation overhead both grow exponentially with the depth of  $U_C$ , thus this framework allows us to trade between these exponentials to optimize the resource requirements and accuracy of quantum hardware experiments. The applicability of our framework thus depends on several factors, including the noise profile of the quantum device and details of the problems at hand, e.g. how close the circuits are to Clifford circuits.

### Experimental demonstration

In this section, we demonstrate the operator backpropagation technique on a Hamiltonian simulation experiment. In particular, we consider the simulation of the XY model with nearest-neighbor couplings

$$H = \sum_{i,j \in \mathcal{E}(\Lambda)} J (X_i X_j + Y_i Y_j) + h \sum_{i \in \Lambda} Z_i, \tag{4}$$

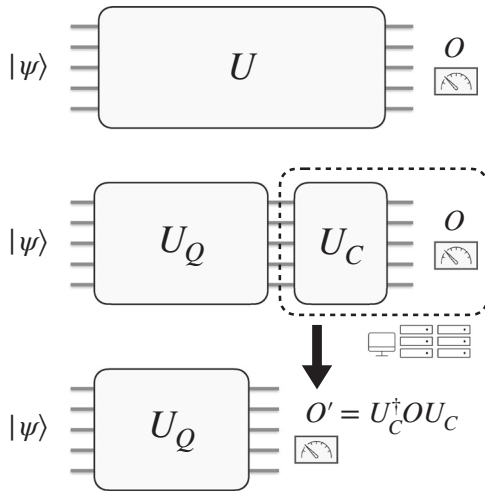
where  $X_i, Y_i, Z_i$  are the Pauli operators supported on site  $i$  and  $\mathcal{E}(\Lambda)$  is the set of edges of a  $D$ -dimensional regular lattice  $\Lambda$ , e.g. a one-dimensional chain or a two-dimensional heavy-hex lattice. For all experiments performed, we consider this model where  $J = 1, h = 0$ , and we are interested in estimating the polarization  $M \equiv \frac{1}{n} \sum_i Z_i$ .

To simulate  $e^{-iHt}$ , we Trotterize the time evolution and approximate it with  $U(\tau)^{t/\tau}$ , where  $\tau$  is the Trotter step size,  $t$  is taken to be an integer multiple of  $\tau$ , and

$$U(\tau) = e^{-i\tau \sum_i Z_i} \prod_{i,j \in \mathcal{E}(\Lambda)} e^{-i\tau (Y_i Y_j + X_i X_j)}. \tag{5}$$

Since this Trotterization preserves the  $U(1)$  symmetry of the XY model, the expectation value of  $M$  is exactly computable and the error in the measured expectation can be used as a proxy for the error of the simulation. Although  $M$  is a conserved quantity of our Trotter circuits, recovering the dynamics of individual  $Z_i$  operators is not classically efficient in general when  $\Lambda$  is two-dimensional. Using OBP we recover these individual expectation values as well and compare against values obtained by a matrix product state (MPS) calculation.

All experiments were error mitigated using a combinations of zero-noise extrapolation (ZNE)<sup>3,4,28,29</sup> via probabilistic error amplification (PEA)<sup>4,30-33</sup> as well as twirled readout error extinction (TRES)<sup>34</sup>. The noise learning and noise amplification procedures used to apply PEA were performed as in Ref.<sup>6</sup>. In the Supplemental Information, we discuss further the details of the error mitigation used.



**Fig. 1 | Operator backpropagation (OBP) framework.** A quantum circuit  $U$  is split into two subcircuits  $U_C$  and  $U_Q$ . A classical simulator computes the Pauli decomposition of  $O' = U_C^\dagger O U_C$ , which is then measured on quantum hardware.

To benchmark the framework, we first apply it to the simulation of the exactly solvable one-dimensional XY model of 75 spins. We observe that as circuit depth increases, experiments leveraging OBP obtain lower error for estimates of polarization when the total number of circuit executions are held constant. In addition, we run a larger experiment of a two-dimensional XY model on heavy-hex lattice of 127 spins. This system is not exactly solvable and other two-dimensional geometries are known to exhibit interesting phenomena such as topological phase transitions<sup>35</sup>. We observe that for all Trotter depths considered, experiments which estimate the polarization using OBP obtain a reduction in error relative to experiments which do not use OBP.

For both the one and two-dimensional spin models, the spins are mapped to a subset of nearest-neighboring qubits on the coupling graph of `ibm_kyiv`, one of IBM Quantum’s superconducting quantum processors. We fix the length of each Trotter step to be  $\tau = 0.05$  and consider the expectation value of the polarization, i.e.

$$\langle M \rangle_k = \langle \psi | U^\dagger(\tau)^k M U(\tau)^k | \psi \rangle, \tag{6}$$

at different numbers of Trotter steps  $k$ , with and without using OBP to reduce the circuit depth by 5 Trotter steps. We chose this value by fixing an  $L_2$  error budget for the classical approximation of  $M$ , fixing a maximum of 10 qubit-wise commuting Pauli groups, and then selecting the largest whole number of Trotter steps that could be backpropagated within these constraints. For the 75 (127) qubit experiments,  $O'$  contained 370 (881) Pauli terms, respectively. See the Supplemental Information for a detailed description of the classical OBP calculations.

For each OBP computation, the polarization  $M \equiv \frac{1}{n} \sum_i Z_i$  is processed for five Trotter steps by backpropagating each  $Z$  operator independently using an  $L_2$  error budget of 0.01 and 0.025 for the one and two dimensional models, respectively. In each case, the truncation error budgets were distributed unevenly across the circuit slices, withholding a majority of the error budget for the final slice. For both models, the set of Paulis supporting all backpropagated observables are collected into 8 qubit-wise commuting groups, each of which is independently evaluated on a QPU.

Additionally, the Pauli operators needed to reconstruct one through four Trotter steps are a subset of those needed to reconstruct five steps. Thus, we can re-use our measurement data to recover the dynamics of both XY models for all  $k \in [0, 25]$ , despite the fact that we only execute circuits on hardware where  $k$  is an integer multiple of 5.

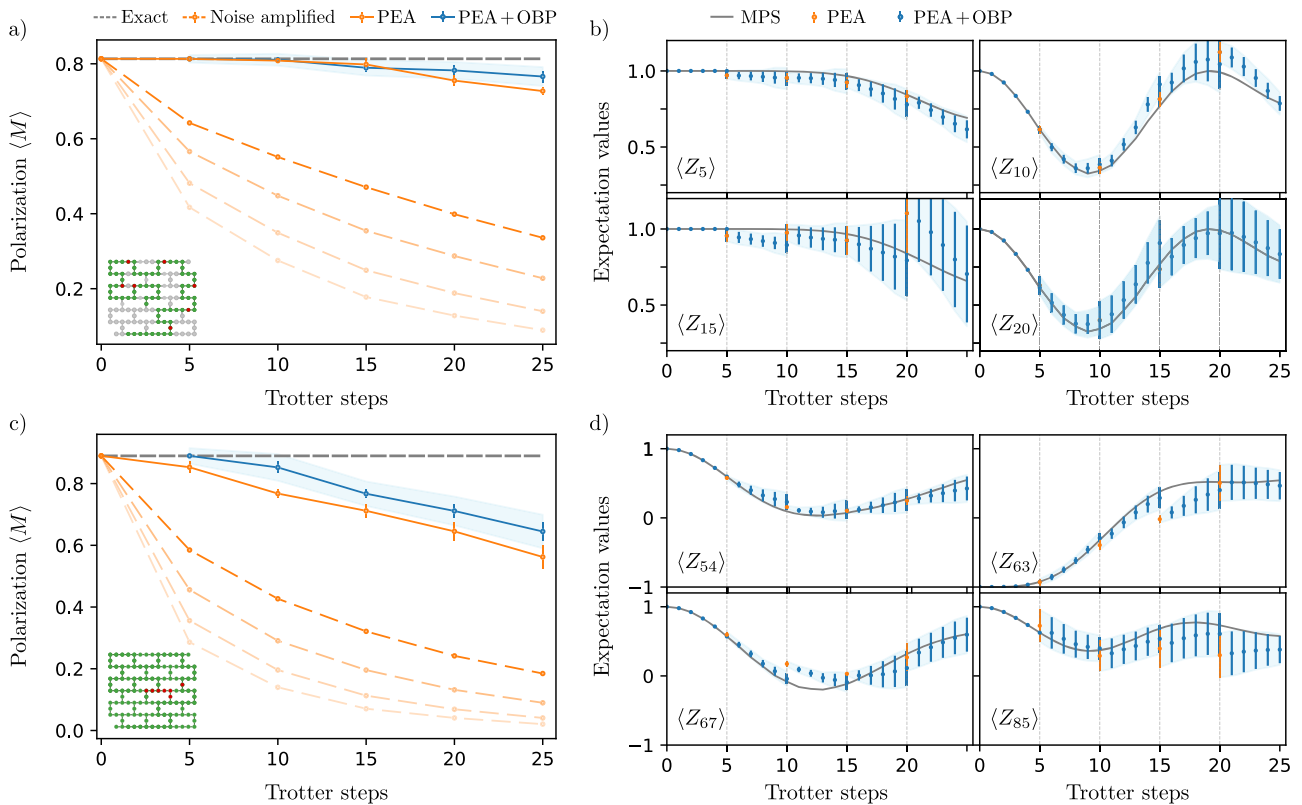
The results shown for all experiments are obtained by bootstrapping the measurements into 100 batches and then taking the mean across all postprocessed batches. The error bars are plotted with a  $2\sigma$  confidence and data which leverages OBP is plotted with a shaded region which includes the statistical error of the experiment as well as the  $L_2$  error budget (See Section III) which was allocated during backpropagation. See the Supplemental Information for more information on the error mitigation used for this experiment.

In Fig. 2a, we plot the expectation value of the polarization, i.e. Eq. (6) for a one dimensional XY model of 75 spins. The deepest circuit executed for this experiment is 25 Trotter steps, which requires 1924 two-qubit gates and a two-qubit circuit depth of 52. The initial state  $|\psi_0\rangle$  is initialized to  $|00\dots 0\rangle$ , except for seven evenly spaced qubits which are initialized to  $|1\rangle$ . Under the dynamics of the XY model, these “excitations” (qubits initialized to  $|1\rangle$ ) will spread to other sites on the lattice. We reconstruct the expectation value of all local  $Z$  operators with and without OBP and compare their mean to the known reference value of  $\frac{75-2*7}{75} \approx 0.813$ . The experiments performed with and without OBP each use a total 262144 circuit executions for every  $k$ . We observe that after 15 Trotter steps, the experiments leveraging OBP achieve a statistically significant reduction in error when estimating  $\langle M \rangle_k$ . Because these experiments use the same number of circuit executions, the estimates obtained via OBP use an overall fewer number of total circuit layer operations - 19% (45%) fewer for the 25 (10) Trotter step data points, respectively. Although the OBP framework will, in general, introduce an overhead in the number of quantum circuit executions (shots), these results highlight that even for a fixed budget of circuit executions, one can reduce the error of a quantum simulation by reducing circuit depth via OBP. Furthermore, because the circuits shortened via OBP have fewer total gates per circuit executions, these experiments require fewer total quantum operations

In Fig. 2b, we highlight another important feature of the OBP framework: the ability to recover the dynamics at intermediate times from coarse measurement data. Specifically, from the measurement after only 5, 10, 15, 20, and 25 Trotter steps, we can use OBP to reconstruct the expectation values of observables at all Trotter steps from 0 to 25. We plot the dynamics of several such observables  $Z_i$  in Fig. 2b and compare against reference values obtained from MPS simulations. The  $Z_i$  displayed are chosen with  $i$  located near the initial excitations (red circles in the inset of Fig. 2a) in order to observe strong dynamical response to the excitations. We note that our estimates of the individual  $Z_i$  dynamics are more impacted by noise than our estimates of  $\langle M \rangle_k$ , which is explained by a concentration about the mean arising from the averaging over all  $Z_i$ . Nevertheless, we observe qualitative agreement between the fine-grained dynamics reconstructed via OBP and those obtained via a MPS simulation. Additionally, we can compare the error mitigated expectation values for the same total number of Trotter steps, but where OBP was or was not used to reduce circuit depth by 5 steps. These data points provide a direct way to compare the impact of OBP for individual observable expectation values.

In addition to the exactly solvable XY spin chain, we also consider a two-dimensional XY model of 127 spins defined on a heavy-hex graph given by the qubit connectivity of the `ibm_kyiv` QPU. The largest circuit executed in this experiment is 25 Trotter steps, which requires 4896 two-qubit gates and a two-qubit depth of 102. Because this model is two-dimensional, the resulting Trotter circuits are approximately twice as deep and contain 254% as many two-qubit gates as the Trotter circuits for the one-dimensional model. These circuits are more strongly impacted by noise, and the light-cones of operator expectation values spread more rapidly across the system. In the Supplemental Information, we detail how the Trotter circuits in this experiment were synthesized.

In Fig. 2c and d, we plot  $\langle M \rangle_k$  for the two-dimensional XY model as well as the expectation values for a handful of local  $Z$  observables, which we compare with reference values obtained through MPS simulations. The MPS calculations are discussed in more detail in the Supplemental Information. Similar to the one-dimensional experiment, we initialize  $|\psi_0\rangle$  to  $|00\dots 0\rangle$  apart from seven excitations which are placed near the center of the



**Fig. 2 | OBP experiments with 75 and 127-qubit spin models.** Benchmarking the OBP framework in the simulation of the one-dimensional XY model of 75 spins (a) and (b) and the two-dimensional XY model of 127 spins (c) and (d). **a/c** Expectation of the polarization  $M$  at different time steps. The polarization is a conserved quantity (dashed line) under the dynamics of the XY model. Due to noise, the experimental signal decays with the depth of the circuit and can be partially recovered using error mitigation. The signals at different noise amplification (by a factor of 1, 1.5, 2.25 and 3, indicated by bolder to more transparent dashed orange lines, respectively) are extrapolated to obtain the PEA estimate (solid orange lines). Using OBP with 5 Trotter steps backpropagated, the polarization can be measured to a higher accuracy in deep circuits (blue lines). The insets highlight the qubits of `ibm_kyiv` used to

represent the spins. The qubits are initialized in either  $|0\rangle$  (green circles) or  $|1\rangle$  (red circles). **b/d** Dynamics of several individual  $Z_i$  under the XY model. The vertical dashed lines indicate Trotter steps at which the expectation values are measured. The orange scatter points indicate results from measurement at 5, 15, and 20 Trotter steps and applying PEA without OBP. The OBP framework helps recover the dynamics of intermediate time values (blue scatter points) from these coarse measurement data. The results agree with the reference values (solid gray lines) obtained via an MPS simulation. All error bars shown were obtained through bootstrapping with 100 batches, and are shown at a  $2\text{-}\sigma$  confidence. The shaded blue region represents the additional  $L_2$  error bound due to the classical approximation of the backpropagated observable.

lattice. We reconstruct the expectation values of all local  $Z$  operators with and without OBP and compare the mean with the known reference value of  $\frac{127-2*7}{127} \approx 0.89$ . For these experiments, each group of qubit-wise commuting Paulis estimated on hardware is allocated 32768 shots, resulting in a total of 262144 (32768) shots for estimates of  $\langle M \rangle_k$  with (without) OBP. We observe that for all number of Trotter steps, estimates of  $\langle M \rangle_k$  which use OBP to reduce the circuit depth by 5 steps obtain a statistically significant decrease in error relative to experiments which do not make use of OBP.

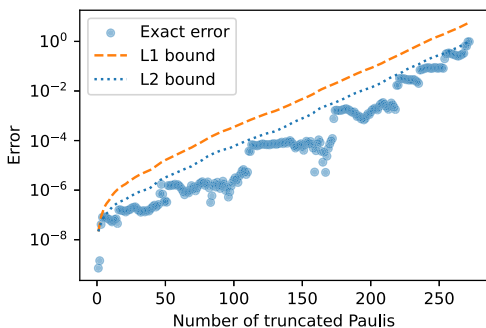
### Discussion

We have introduced the OBP framework for improving the quantum computation of expectation values for Pauli observables on pre-fault tolerant hardware and demonstrated on quantum hardware how OBP can reduce the error of a Hamiltonian time dynamics experiment. This framework reduces the depth of quantum circuits run on quantum hardware, limiting the impact of errors and the cost of error mitigation in exchange for an increase in the number of experiments and an additional classical overhead. In the context of Hamiltonian simulation, OBP reduces the error in approximating the dynamics of quantum systems, allowing longer-time simulation compared to purely quantum approaches.

An important feature of the OBP framework is the ability to reconstruct dynamics at intermediate times from coarse measurement data. For example, in our experiment, we only measure the state after 5, 10, 15, 20, and 25 Trotter steps, but the expectation value of the observable can be

constructed for all Trotter steps between 0 and 25. Additionally, the same expectation value can be computed in multiple ways: for example, the expectation value after 15 Trotter steps can be obtained from the experiment which executes 15 Trotter steps on a QPU and does not backpropagate the target observable, or from the experiment which runs 10 Trotter steps on a QPU and backpropagates the target observable through 5 Trotter steps. In principle, these multiple ways to estimate the same quantity, each suffering from a different set of errors, may be used to obtain a more accurate estimate of the expectation value. Exploring such an error-mitigation scheme is an interesting future direction.

The OBP framework is convenient whenever measuring the backpropagated observable requires a manageable number of quantum circuit executions. Generally, this requirement puts a constraint on the depth of the circuit  $U_C$  through which an observable can be backpropagated. If one fixes an error budget and a maximum overhead in the number of observable measurements, then the reduction in circuit depth which can be achieved will depend strongly on the circuit which one attempts to backpropagate. The OBP framework will show the most promise for problems where the size of one's backpropagated observable grows slowly with the depth of  $U_C$ . This can be achieved in situations where, for example,  $U_C$  consists of few non-Clifford operations or the dynamics of a quantum systems experience periodicity at stroboscopic times, at which point the backpropagated observable may grow slowly enough that one can obtain substantial depth reduction before the incurred overhead grows prohibitively large.



**Fig. 3 | Comparison between  $L_1$  and  $L_2$  error bounds vs. exact error of OBP truncation.** The exact error  $\|\Delta\|$  from truncating a backpropagated observable (blue circles) and the error bounds based on the  $L_1$  norm [Eq. (8), orange dashed line] and the  $L_2$  norm [Eq. (10), blue dotted line] of the truncated coefficients. Here,  $U_Q$  and  $U_C$  each correspond to 5 Trotter step time evolution circuits for a 12 qubit XY model [Eq. (5)] on a one-dimensional lattice with closed boundary conditions at Trotter step size  $dt = 0.1$ . The initial state is  $|00 \dots 0\rangle$  and the target observable is  $Z_1$ . The total number of Pauli terms in the backpropagated observable without truncation is 271.

Identifying such use cases where backpropagation yields substantial depth reduction while the full experiment remains classically hard to simulate is an interesting direction for future work.

## Methods

### Truncation error

Recall that the backpropagated observable  $O'$  is written as a linear combination of multi-qubit Pauli operators (cf. Eq. (2)). Some coefficients  $c_P$  in the decomposition may be small enough that they can be truncated from  $O'$  without incurring significant error. We discuss the estimation of this truncation error in this section.

Suppose that  $\mathcal{S}$  is the set of Paulis upon which  $O'$  is supported,  $\mathcal{T}$  is the subset of Paulis which will be truncated, and  $\mathcal{K}$  is the subset of the remaining Paulis which will remain after truncation. Let  $O'_K = \sum_{P \in \mathcal{K}} c_P P$  be the truncated version of  $O'$ . The difference between them is

$$\Delta \equiv O' - O'_K = \sum_{P \in \mathcal{T}} c_P P. \tag{7}$$

Using the triangle inequality, the truncation error can be bounded by the  $L_1$  norm of the truncated coefficients,

$$|\langle \psi_Q | \Delta | \psi_Q \rangle| \leq \sum_{P \in \mathcal{T}} |c_P|, \tag{8}$$

where  $|\psi_Q\rangle = U_Q |\psi\rangle$ . This error bound is rigorous, but it can only be saturated if the truncated Pauli operators mutually commute and  $|\psi_Q\rangle$  happens to be a common eigenstate of the operators. Except for these worst cases, the  $L_1$  norm of the truncated coefficients will largely overestimate the truncation error.

Instead, we may use a different estimate that is expected to better capture the truncation error in typical cases. To motivate the estimate, we first assume that  $|\psi_Q\rangle$  were drawn from a 1-design ensemble. The error in the expectation value follows a distribution with a vanishing mean and a variance given by<sup>24</sup>

$$\mathbb{E}_{|\psi_Q\rangle} |\langle \psi_Q | \Delta | \psi_Q \rangle|^2 \leq \frac{1}{2^n} \text{Tr}(\Delta^2) = \sum_{P \in \mathcal{T}} |c_P|^2. \tag{9}$$

Thus, the majority of the drawn  $|\psi_Q\rangle$  would result in a truncation error smaller than the  $L_2$  norm of the truncated coefficients. In our problem,  $|\psi_Q\rangle$  is deterministic instead of being randomly drawn from an ensemble. But if  $U_Q$  is a generic and sufficiently deep circuit, we expect Eq. (9) better captures

the typical values of  $\langle \psi_Q | \Delta | \psi_Q \rangle$  than the worst-case bound [Eq. (8)], leading to an estimate

$$|\langle \psi_Q | \Delta | \psi_Q \rangle| \lesssim \left( \sum_{P \in \mathcal{T}} |c_P|^2 \right)^{1/2}. \tag{10}$$

We benchmark both error bounds in Fig. 3 and show that it is indeed the case for even a relatively shallow circuit.

### Operator backpropagation via Clifford perturbation theory

Our framework requires the decomposition of the backpropagated observable  $O'$  in the computational basis. In this section, we discuss an implementation of such an OBP algorithm based on the Clifford perturbation theory (CPT)<sup>21</sup>. The key idea is that Clifford circuits can be efficiently simulated by classical computers and we can realize an algorithm using CPT whose complexity scales exponentially with the non-Cliffordness of  $U_C$ .

The OBP algorithm based on CPT works as follows. First, we split the subcircuit to be backpropagated,  $U_C$ , into slices,

$$U_C = \prod_{s=1}^S U_s \equiv U_S \dots U_2 U_1, \tag{11}$$

where  $S$  denotes the total number of slices and  $U_s$  represents a single slice of  $U_C$ . No constraints are made on the depth of the slices and each slice may be supported on all qubits. The purpose of splitting of  $U_C$  into slices is to provide timely stopping points within the OBP algorithm to control the growth of the intermediate observable  $O^{(s)}$  by truncating terms with low magnitude coefficients. The algorithm proceeds by iterating over the circuit slices *in reverse*, i.e. starting from slice  $S$  and ending on slice 1. For each slice, all quantum gates are analytically applied to the current operator,

$$O^{(s)} = U_{S-s+1}^\dagger O^{(s-1)} U_{S-s+1}, \tag{12}$$

where  $s$  denotes the iteration index of the OBP algorithm and  $O^{(0)} = O$  is the original operator whose expectation value we are interested in.

For example, if  $O = X_1$  is a Pauli  $X$  supported on qubit 1 and  $U_s$  is a CNOT gate between qubits 1 and 2,  $O^{(1)} = U_s^\dagger X_1 U_s$  again contains only one Pauli string  $X_1 X_2$ . However, if  $U_s$  were a  $T$  gate on qubit 1,  $O^{(1)}$  would instead contain two Pauli operators:  $X_1$  and  $Y_1$ . In general, the number of Paulis in the backpropagated observable remains the same after applying a Clifford gate and may double after each non-Clifford one. Therefore, during the course of the OBP algorithm, the number of Pauli terms comprising  $O^{(s)}$  can grow exponentially.

The OBP algorithm based upon CPT is particularly convenient when  $U_C$  contains gates that are unitary rotations by small angles. Such rotations often arise in a Trotterized Hamiltonian simulation. In such cases, many of the coefficients in the decomposition of  $O^{(s)}$  are small, providing an opportunity to truncate them from the operator. In our implementation, truncation occurs after the backpropagation of each slice and our total error budget is divided up among the fixed number of slices.

At this point we emphasize once more that the slices are not restricted to be of depth 1. For example, when a  $X_1 X_2$  and  $Y_1 Y_2$  rotation are applied subsequently on the same pair of qubits 1 and 2, truncating terms between their individual backpropagation may not be desirable, because backpropagating both gates at once may itself result in beneficial cancellation of terms. This choice might be made, a-priori, if one knows that the operator being backpropagated, e.g.  $Z_1 + Z_2$ , is an approximate symmetry of an  $X_1 X_2 + Y_1 Y_2$  rotation. This is precisely the situation encountered in the experiments presented in Section II and discussed further in the Supplemental Information, where slices are chosen to contain a single layer of  $XX + YY$  rotations.

Truncation of the operator reduces the memory required to store its Pauli decomposition, but it also leads to a challenge in tracking the total truncation error. Because truncation occurs at different points in the

Paulis	Binary Encoding		Address	Nodes
	Z	X		
IIII	0000	0000	0	node 1: 0..63
IZIZ	0101	0000	80	node 2: 64..127
IXXI	0000	0110	6	node 3: 128..191
YZIX	1100	1001	201	node 4: 192..255
⋮	⋮	⋮	⋮	

**Fig. 4 | Example of distributing Pauli terms based on their binary encoding index.** From left to right we show how a Pauli term is encoded as a binary array which then gets interpreted as an address that can be mapped into the address range associated with a given node.

Heisenberg evolution, the orthogonality of truncated components cannot be assumed. Thus one may not use Eq. (8) or Eq. (10) directly. Instead, one can use the triangle inequality to further upper bound the total truncation error

$$\epsilon \leq \epsilon_1 + \epsilon_2 + \dots + \epsilon_s, \tag{13}$$

where  $\epsilon_s$  is the truncation error, evaluated using either Eq. (8) or Eq. (10), at the  $s$ th iteration.

For the implementation of OBP discussed here, a total error budget is specified up front which is divided among each slice of  $U_C$ . At each truncation step, Pauli terms are removed in order of increasing coefficient magnitude until no further terms can be truncated without exceeding the *per-slice* error budget. At the end of truncation, any residual error budget is added to the *per-slice* error budget of the next slice. The complexity of this procedure scales as  $\mathcal{O}(|\mathcal{S}_s| \log |\mathcal{S}_s|)$  where  $\mathcal{S}_s$  is the set of Pauli terms comprising  $O^{(s)}$  prior to truncation.

The implementation of OBP described above will terminate if any of the three following conditions are met: the observable  $O$  has been backpropagated through all circuit slices,  $U_s$ ; after expending the truncation budget, the size of  $\mathcal{K}$  is over the user-specified limit; or the algorithm exceeded a user-specified runtime limit.

The latter two early-termination criteria are motivated by practical constraints. The total number of Paulis in the decomposition of  $O$ , i.e.  $|\mathcal{K}|$ , is a proxy for both the classical memory requirements as well as the total number of quantum circuits which need to be executed on hardware. It is also possible to terminate based on the minimal number of qubit-wise-commuting Pauli groups, which defines the true number of required circuit executions. However, computing this quantity is NP-Hard<sup>36</sup> and approximating it for large  $|\mathcal{K}|$  can be prohibitively expensive.

### Parallelization of operator backpropagation

Backpropagation of observables is a form of simulation and thus has exponential overhead in general. Employing truncation schemes can extend the reach of this method, however, it is natural to question how well these classical algorithms can be scaled for large problem instances. In this section we discuss the challenges of scaling our method to a highly parallelized architecture like one might see in an HPC cluster. We observe that one immediate obstacle lies in the potentially prohibitive amount of all-to-all communication between computational nodes which arises from a naive parallelization of the norm-based truncation strategy we describe in Section III. Finally, we describe a scheme for distributing the OBP calculation across nodes where the number of messages passed scales only linearly in the total number of nodes.

Multiple approaches can be taken to parallelize the implementation of the OBP algorithm. The most naive approach would be to simply parallelize the backpropagation of multiple operators of interest but this assumes that one indeed has multiple operators to begin with. Furthermore, this does not resolve the fundamental limitation imposed by memory being the most likely resource constraint for how many circuit slices can be backpropagated. Therefore, a multi-node parallelization for the backpropagation of any single operator is desirable.

The OBP implementation based on CPT poses two major difficulties for such a parallelization. First, when a slice of circuit operations gets backpropagated, each operation has to be applied to each Pauli term. If the circuit operation commutes with the Pauli term, this is trivial. Otherwise, if the operation is non-Clifford, the backpropagation results in up to twice the number of Pauli terms, possibly leading to new terms not present previously. When this is done in parallel on disjoint sets of Pauli terms, the resulting sets may no longer be disjoint. If one now desires to perform low-magnitude coefficient truncation while adhering to an error budget, the sets of Pauli terms must be de-duplicated such that coefficients of duplicate Pauli terms are properly summed. Naively, this would involve either an all-to-all or one-to-all communication pattern, posing a severe bottleneck on the parallelization efficiency, and in the latter case, bottlenecking the entire computation by any single node’s memory capacity. Second, agreeing on a truncation threshold over a distributed set of Pauli terms also poses a potential difficulty. In the following, we are going to propose a solution to overcome both of these limitations.

The first problem can be tackled by realizing that all  $4^N$  Pauli terms that span the space of  $N$  qubits have a natural ordering. Every Pauli term can be represented via two bitstrings of length  $N$  where each bitstring encodes the presence (1) or lack (0) of a Z or X Pauli at a given index<sup>37–39</sup>. This is possible due to the relations of the identity and three Pauli matrices. Therefore, in this representation, the four possible combinations of two bits encode the following:  $(0, 0) \rightarrow I$ ,  $(0, 1) \rightarrow Z$ ,  $(1, 0) \rightarrow X$ ,  $(1, 1) \rightarrow Y$ .

Further, we can concatenate the bitstrings encoding the Z and X Paulis to form a single bitstring of length  $2N$ . This bitstring serves as a unique identifier for every possible Pauli term and provides a natural order to the entire set. Thus, given any Pauli term, its bitstring can be used as an address to quickly determine its position in the larger set.

When distributing Pauli terms across  $R$  compute nodes, the entire address space can be partitioned into  $R$  intervals, associated to the different nodes. This distribution scheme ensures that for any Pauli address, the corresponding node whose interval contains this Pauli can be determined in constant time. This procedure is visualized in Fig. 4 for a specific case involving 4 qubits and a selection of Pauli terms which are to be assigned to one of the four compute nodes. From left to right, the figure shows how a Pauli gets encoded in the ZX calculus which then gets interpreted as an address that can be mapped into the address range associated with a given node.

The scheme presented above addresses the challenge of distributed deduplication and requires each node to exchange at most  $R$  messages. In the Supplemental Information, we further show how the partitioning of Pauli addresses can be efficiently updated in order to balance the number of Paulis within each node’s interval, requiring  $\mathcal{O}(R)$  total messages to be passed.

Now that a distributed storage system for the Pauli terms has been devised, backpropagation of any circuit slice can be performed in parallel on disjoint sets of Pauli terms. If we wish to truncate Pauli terms with low-magnitude coefficients, this can also be done in parallel but care must be taken not to exceed any specified error budget. To this end, every node can in parallel determine the smallest and largest coefficient magnitude and communicate these values to a master node. Upon receiving all lower and upper bounds, the master node can propose a truncation threshold and broadcast this to all nodes. Each node can then compute its own truncation error for the proposed threshold and send this information back to the main rank. By iterating on this procedure, a binary search for an agreeable truncation threshold can be performed on a distributed set of Pauli terms and coefficients. Therefore,  $\mathcal{O}(R \log(|\mathcal{S}|))$  broadcast messages are required to determine the truncation threshold, where  $\mathcal{S}$  is the set of Paulis upon which the backpropagated observable is supported, prior to truncation.

*Note added:* During the preparation of this manuscript, we learned of complementary works by Lerch et al.<sup>40</sup>, Rudolph et al.<sup>41</sup>, Miller et al.<sup>42</sup>, and Faehrmann et al.<sup>43–42</sup>, each present techniques similar to OBP where operators are propagated backward through entire circuits as a form of simulation. These manuscripts discuss valuable ideas outside the

scope of this work including different approaches to operator truncation, extensions to fermionic systems, and variations where the coefficients of backpropagated operators are represented symbolically as a function gate rotation angles. Meanwhile, Faehrmann et al. proposed to backpropagate an observable through a short-time evolution of a Hamiltonian using the truncated Taylor series. In general, one can instead construct a Trotterized approximation to the dynamics and apply the OBP framework detailed in this manuscript. Comparing the efficiency of the two approaches and identifying applications where one can be more beneficial than the other is of great practical interest, but is outside the scope of the current manuscript.

### Data availability

Data generated and analyzed for this manuscript are available from the corresponding author upon reasonable request. Code used to generate data in this manuscript are available from the corresponding author upon reasonable request.

### Code availability

Code used to generate data in this manuscript are available from the corresponding author upon reasonable request.

Received: 26 March 2025; Accepted: 2 February 2026;

Published online: 19 February 2026

### References

- Lloyd, S. Universal quantum simulators. *Science* **273**, 1073–1078 (1996).
- Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
- Temme, K., Bravyi, S. & Gambetta, J. M. Error mitigation for short-depth quantum circuits. *Phys. Rev. Lett.* **119**, 180509 (2017).
- Li, Y. & Benjamin, S. C. Efficient variational quantum simulator incorporating active error minimization. *Phys. Rev. X* **7**, 021050 (2017).
- Cai, Z. et al. Quantum error mitigation. *Rev. Mod. Phys.* **95**, 045005 (2023).
- Kim, Y. et al. Evidence for the utility of quantum computing before fault tolerance. *Nature* **618**, 500–505 (2023).
- Nation, P. D. & Treinish, M. Suppressing quantum circuit errors due to system variability. *PRX Quantum* **4**, 010327 (2023).
- Kremer, D. et al. Practical and efficient quantum circuit synthesis and transpiling with reinforcement learning. *arXiv:2405.13196* <https://doi.org/10.48550/arXiv.2405.13196> (2024).
- Filippov, S., Leahy, M., Rossi, M. A. C. & García-Pérez, G. Scalable tensor-network error mitigation for near-term quantum computing 2307.11740 (2023).
- Vazquez, A. C., Egger, D. J., Ochsner, D. & Woerner, S. Well-conditioned multi-product formulas for hardware-friendly hamiltonian simulation. *Quantum* **7**, 1067 (2023).
- Robertson, N. F. et al. Tensor network enhanced dynamic multiproduct formulas <https://arxiv.org/abs/2407.17405>. 2407.17405 (2024).
- Robertson, N. F., Akhriev, A., Vala, J. & Zhuk, S. Approximate quantum compiling for quantum simulation: A tensor network based approach. *arXiv:2301.08609* <https://arxiv.org/abs/2301.08609> (2023).
- Madden, L. & Simonetto, A. Best approximate quantum compiling problems. *ACM Trans. Quantum Comput.* **3** <https://doi.org/10.1145/3505181> (2022).
- Rudolph, M. S., Chen, J., Miller, J., Acharya, A. & Perdomo-Ortiz, A. Decomposition of matrix product states into shallow quantum circuits. *Quantum Sci. Technol.* **9**, 015012 (2023).
- Ben-Dov, M., Shnaiderov, D., Makmal, A. & Dalla Torre, E. G. Approximate encoding of quantum states using shallow circuits. *npj Quantum Inf.* **10** <https://doi.org/10.1038/s41534-024-00858-1> (2024).
- Lin, S.-H., Dilip, R., Green, A. G., Smith, A. & Pollmann, F. Real- and imaginary-time evolution with compressed quantum circuits. *PRX Quantum* **2**, 010342 (2021).
- Ran, S.-J. Encoding of matrix product states into quantum circuits of one- and two-qubit gates. *Phys. Rev. A* **101**, 032310 (2020).
- Carrera Vazquez, A. et al. Combining quantum processors with real-time classical communication. *Nature* **636**, 75–79 (2024).
- Krishnan Vijayan, M. et al. Compilation of algorithm-specific graph states for quantum circuits. *Quantum Science and Technology* **9**, 025005 (2024).
- Kaldenbach, T. N. & Heller, M. Mapping quantum circuits to shallow-depth measurement patterns based on graph states. *Quantum Science and Technology* **10**, 015010 (2024).
- Begušić, T., Hejazi, K. & Chan, G. K.-L. Simulating quantum circuit expectation values by clifford perturbation theory. <http://arxiv.org/abs/2306.04797v2>. 2306.04797v2 (2023).
- Fuller, B. et al. Qiskit addon: Operator Backpropagation. <https://github.com/Qiskit/qiskit-addon-obp> (2024).
- Begušić, T., Gray, J. & Chan, G. K.-L. Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance. *Sci. Adv.* **10** <https://doi.org/10.1126/sciadv.adk4321> (2024).
- Schuster, T., Yin, C., Gao, X. & Yao, N. Y. A polynomial-time classical algorithm for noisy quantum circuits. <https://arxiv.org/abs/2407.12768>. 2407.12768 (2024).
- Angrisani, A. et al. Classically estimating observables of noiseless quantum circuits <http://arxiv.org/abs/2409.01706>. arXiv:2409.01706 (2024).
- Alexeev, Y. et al. Quantum-centric supercomputing for materials science: A perspective on challenges and future directions. *arXiv preprint arXiv:2312.09733* (2023).
- Tindall, J., Fishman, M., Stoudenmire, E. M. & Sels, D. Efficient tensor network simulation of ibm's eagle kicked ising experiment. *PRX Quantum* **5** <https://doi.org/10.1103/PRXQuantum.5.010308> (2024).
- Kim, Y. et al. Scalable error mitigation for noisy quantum circuits produces competitive expectation values. *Nat. Phys.* **19**, 752–759 (2023).
- Kandala, A. et al. Error mitigation extends the computational reach of a noisy quantum processor. *Nature* **567**, 491–495 (2019).
- McDonough, B. et al. Automated quantum error mitigation based on probabilistic error reduction. In *2022 IEEE/ACM Third International Workshop on Quantum Computing Software (QCS)*, 83–93 <https://ieeexplore.ieee.org/document/10025519> (2022).
- Mari, A., Shammah, N. & Zeng, W. J. Extending quantum probabilistic error cancellation by noise scaling. *Phys. Rev. A* **104**, 052607 (2021).
- Ferracin, S. et al. Efficiently improving the performance of noisy quantum computers. *Quantum* **8**, 1410 (2024).
- Endo, S., Benjamin, S. C. & Li, Y. Practical quantum error mitigation for near-future applications. *Physical Review X* **8** <https://doi.org/10.1103/PhysRevX.8.031027> (2018).
- van den Berg, E., Mineev, Z. K. & Temme, K. Model-free readout-error mitigation for quantum expectation values. *Phys. Rev. A* **105**, 032620 (2022).
- Ding, H.-Q. Phase transition and thermodynamics of quantum xy model in two dimensions. *Phys. Rev. B* **45**, 230–242 (1992).
- Yen, T.-C., Verteletskyi, V. & Izmaylov, A. F. Measuring all compatible operators in one series of single-qubit measurements using unitary transformations. *J. Chem. Theory Comput.* **16**, 2400–2409 (2020).
- Coecke, B. & Duncan, R. Interacting quantum observables. In Aceto, L. et al. (eds.) *Automata, Languages and Programming*, 298–310 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008). <https://doi.org/10.1007/978-3-540-70583-3>.
- Coecke, B. & Duncan, R. Interacting quantum observables: categorical algebra and diagrammatics. *New J. Phys.* **13**, 43016 (2011).
- Aaronson, S. & Gottesman, D. Improved simulation of stabilizer circuits. *Phys. Rev. A* **70**, 052328 (2004).

40. Lerch, S. et al. Efficient quantum-enhanced classical simulation for patches of quantum landscapes <https://arxiv.org/abs/2411.19896> (2024). 2411.19896.
41. Rudolph, M. S., Jones, T., Teng, Y., Angrisani, A. & Holmes, Z. Pauli propagation: A computational framework for simulating quantum systems <https://arxiv.org/abs/2505.21606> (2025). 2505.21606.
42. Miller, A. et al. Simulation of fermionic circuits using majorana propagation <https://arxiv.org/abs/2503.18939> (2025). 2503.18939.
43. Faehrmann, P. K., Eisert, J., Kieferova, M. & Kueng, R. Short-time simulation of quantum dynamics by Pauli measurements <http://arxiv.org/abs/2412.08719> (2024). ArXiv:2412.08719 [quant-ph].

## Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, and National Quantum Information Science Research Centers. Y.A. and D.L. acknowledge support from the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357 at Argonne National Laboratory. We would also like to acknowledge informative discussions with Petar Jurcevic, Oles Shtanko, Bibek Pokharel, Andrew Eddins, Zlatko Mineev, Conrad Haupt, and Daniel Egger.

## Author contributions

M.T., C.J., M.R., B.F. implemented the classical CPT-based simulation tools, wrote the main manuscript text, and made all figures in the main text. B.F., Y.K., A.H., K.W., A.K. executed the quantum hardware experiments. M.T., K.S., A.M., B.F. conceived of the original project idea. All authors contributed to editing the final manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41534-026-01196-0>.

**Correspondence** and requests for materials should be addressed to Bryce Fuller.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026