PRX QUANTUM **6,** 020360 (2025)

Tensor Network Enhanced Dynamic Multiproduct Formulas

Niall F. Robertson,[1,*,†] Bibek Pokharel,[2,*] Bryce Fuller,[3] Eric Switzer,[4,5,6] Oles Shtanko,[2]
Mirko Amico,[3] Adam Byrne,[1] Andrea D'Urbano,[1] Salome Hayes-Shuptar,[1] Albert Akhriev,[1]
Nathan Keenan,[1] Sergey Bravyi,[3] and Sergiy Zhuk[1]

[1] *IBM Quantum, IBM Research Europe—Dublin, IBM Technology Campus, Dublin 15, Ireland*
[2] *IBM Quantum, IBM Research—Almaden, San Jose California, 95120, USA*
[3] *IBM Quantum, IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598, USA*
[4] *Donostia International Physics Center (DIPC), 20018 Donostia–San Sebastian, Euskadi, Spain*
[5] *Department of Physics, University of Central Florida, Orlando, Florida 32816, USA*
[6] *Nanoscale Device Characterization Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA*

(Received 10 October 2024; accepted 29 May 2025; published 27 June 2025)

Tensor networks and quantum computation are two of the most powerful tools for the simulation of quantum many-body systems. Rather than viewing them as competing approaches, here we consider how these two methods can work in tandem. We introduce a novel algorithm that combines tensor networks and quantum computation to produce results that are more accurate than what could be achieved by either method used in isolation. Our algorithm is based on multiproduct formulas (MPFs)—a technique that linearly combines Trotter product formulas to reduce algorithmic error. It uses a quantum computer to calculate the expectation values and tensor networks to calculate the coefficients used in the linear combination. We present a detailed error analysis of the algorithm and demonstrate the full workflow on a one-dimensional quantum simulation problem on 50 qubits using two IBM quantum computers, IBM_TORINO and IBM_KYIV.

DOI: 10.1103/8bzc-dlgt

I. INTRODUCTION

Understanding out-of-equilibrium properties of quantum systems relevant for chemistry, material science, and high-energy physics often requires the simulation of Hamiltonian dynamics. For example, correlation functions describing unitary time evolution of interacting quantum spins or electrons provide information about the excitation spectrum and aid identification of exotic quasiparticles such as unpaired Majorana fermions in one-dimensional (1D) [1] and two-dimensional (2D) [2] models. Correlation functions combining forward and backward time evolution known as out-of-time-order correlators are commonly used for diagnosing quantum chaos in many-body systems [3], with applications ranging from superconductivity [4] to black-hole physics [5,6].

The ability of conventional classical computers to simulate Hamiltonian dynamics is limited by the exponential cost of representing entangled quantum states. Consider the simplest version of the problem—simulating the dynamics of a 1D spin chain with short-range interactions starting from an initial unentangled state. The entanglement entropy between the left and right halves of the chain typically grows linearly with the evolution time t, until it saturates at the value proportional to the system size n (see, e.g., Ref. [7]). Most classical simulation methods rely on matrix product states (MPSs) to approximate the time-evolved states. Since the bond dimension of MPSs grows exponentially with the amount of entanglement, the simulation cost grows exponentially with n or t. This limits the applicability of classical simulators to small systems or short evolution times.

In contrast, quantum computers can efficiently simulate Hamiltonian dynamics for most practically relevant Hamiltonians [8], at least in theory. For example, quantum algorithms based on high-order Trotter formulas [9] or the Lieb-Robinson bound [10] can simulate the dynamics of spin-chain Hamiltonians with the gate complexity

---

[*] Co-first author. These two authors contributed equally to this work

[†] Contact author: niall.robertson@ibm.com

scaling almost linearly with the space-time volume nt. It is expected that simulation problems of this type with space-time volume $nt \approx 10^4$ are already intractable for existing classical computers [11].

It should be emphasized that even quantum computers cannot simulate Hamiltonian dynamics exactly. To begin with, the existing quantum processors are not fault tolerant and the accumulation of errors limits the depth of quantum circuits that can be executed reliably. Even leaving the fault-tolerance problem aside, a quantum computer can only approximate the exact time-evolution unitary due to algorithmic errors known as Trotter errors [9]. Although the asymptotic run-time of the best known quantum algorithms scales only logarithmically with the desired error tolerance [12], the cost of high-precision simulations is prohibitive for near-term quantum processors. For example, the benchmark problem of Ref. [11] with space-time volume $nt \approx 10^4$ would require nearly 10^7 controlled-NOT (CNOT) gates to approximate the time evolution within three digits of precision [13,14]. This exceeds the size of quantum circuits demonstrated to date by many orders of magnitude.

A natural question is whether classical and quantum simulation algorithms working in tandem can accomplish Hamiltonian-dynamics simulation at a lower cost compared with the classical or quantum algorithms alone. Here, we begin to address this question by showing how to combine classical tensor-network algorithms for simulating weakly entangling quantum circuits [15] with the quantum simulation algorithms based on multiproduct formulas (MPFs) [16–20]. MPFs are designed to produce an expectation value of an observable quantity $\langle O\rangle$ by taking a linear combination of approximate expectation values $\sum c_i\langle O_i\rangle$ obtained from multiple different Trotterization formulas, such that the sum more closely represents the expectation value $\langle O\rangle$ corresponding to the exact continuous-time evolution of the system than any of the individual values $\langle O_i\rangle$.

Previous work [20] has considered a variant of standard MPFs where the coefficients in the linear combination depend on the simulation time t such that $c_i = c_i(t)$. The tensor network enhanced MPF algorithm introduced in this work uses matrix product operators (MPOs), we henceforth refer to it as MPO-MPF. The full MPO-MPF method is illustrated in Fig. 1. The MPO-MPF algorithm is similar to the dynamic MPF algorithm in Ref. [20] in that it also produces coefficients $c_i(t)$ that depend on time. However, there are a number of core differences between MPO-MPF and the previous work that we now outline. First, the coefficients $c_i(t)$ in MPO-MPF are calculated *classically*, unlike the algorithm in Ref. [20]. In this paradigm, the errors in $c_i(t)$ are thus completely controlled by the classical resources available rather than by the noise on the quantum device. Second, the MPO-MPF coefficients are calculated independently at each time t—there is thus no propagation of errors with time, as was the case in the work in Ref. [20]. Furthermore, the MPO-MPF algorithm produces error bounds and can thus identify the regime of time for which it is advantageous to apply MPFs—this is referred to as the "MPF test" and the "Trotter test" in Sec. IV. We also point out that the MPO-MPF algorithm can be applied in combination with previously introduced circuit-optimization algorithms [21,22]—i.e., one can use it to take linear combinations of optimized circuits to further reduce Trotter error (see the discussion in Sec. V and Appendix C). Finally, we note that the quantum resources required to implement the MPO-MPF algorithm are orders of magnitude smaller than the work in Ref. [20], thus allowing us to present some initial results of the MPO-MPF algorithm from a 50-qubit experiment on a real quantum computer.

To illustrate how we use MPOs as part of our dynamic MPF workflow, consider the following problem: given some target state $|\psi\rangle$, a set of basis states $|\psi_j\rangle$ and their corresponding pure-state density matrices ρ and ρ_j, what coefficients c_j optimally project the target density matrix onto the span of the basis-state density matrices ρ_j? Specifically, we want to minimize the Frobenius-norm distance between the target density matrix and some linear combination of the basis states. To calculate the ideal projection coefficients c_j in this linear combination, one only needs the quantities $|\langle\psi_j|\psi\rangle|^2$ and $|\langle\psi_i|\psi_j\rangle|^2$. For the case of interest here, we take the target state $|\psi\rangle$ to be the exactly evolved state $e^{-itH}|\psi_0\rangle$, where $|\psi_0\rangle$ is an initial weakly entangled state, and the basis states forming the span to be the Trotter-evolved states $S(t/k_j)^{k_j}|\psi_0\rangle$, where the k_j are the number of Trotter steps used to produce each basis state ρ_j. When using a purely classical approach to calculate expectation values $\langle\psi|O|\psi\rangle$, one can use tensor network based algorithms such as time-evolving block decimation (TEBD) [23], the time-dependent variational principle (TDVP) [24,25], or $W^{I,II}$ [26] to first prepare the state $|\psi\rangle$ as an MPS, followed by a contraction with the operator O and the conjugated state $\langle\psi|$ to obtain the result. Alternatively, one can evolve the operator to get an MPO approximation for $U^\dagger OU$ and contract this object with the weakly entangled state $|\psi_0\rangle$. This latter approach can be advantageous due to partial cancellation of the unitaries—note that the MPO would be trivial if O was the identity operator (for a review, see Ref. [27]). As discussed in Ref. [27], the difference between these two methods essentially boils down to a different choice of the contraction path. More recent work has taken advantage of this latter choice of contraction path for the simulation of Lindblad evolution operators as part of an error-mitigation workflow [28], where this idea is described as a "middle out contraction." Here, we use MPOs to obtain an approximation of the fidelities between *pure* states for the purpose of obtaining the optimal projection coefficients c_j as part of our dynamic MPO-MPF workflow. More

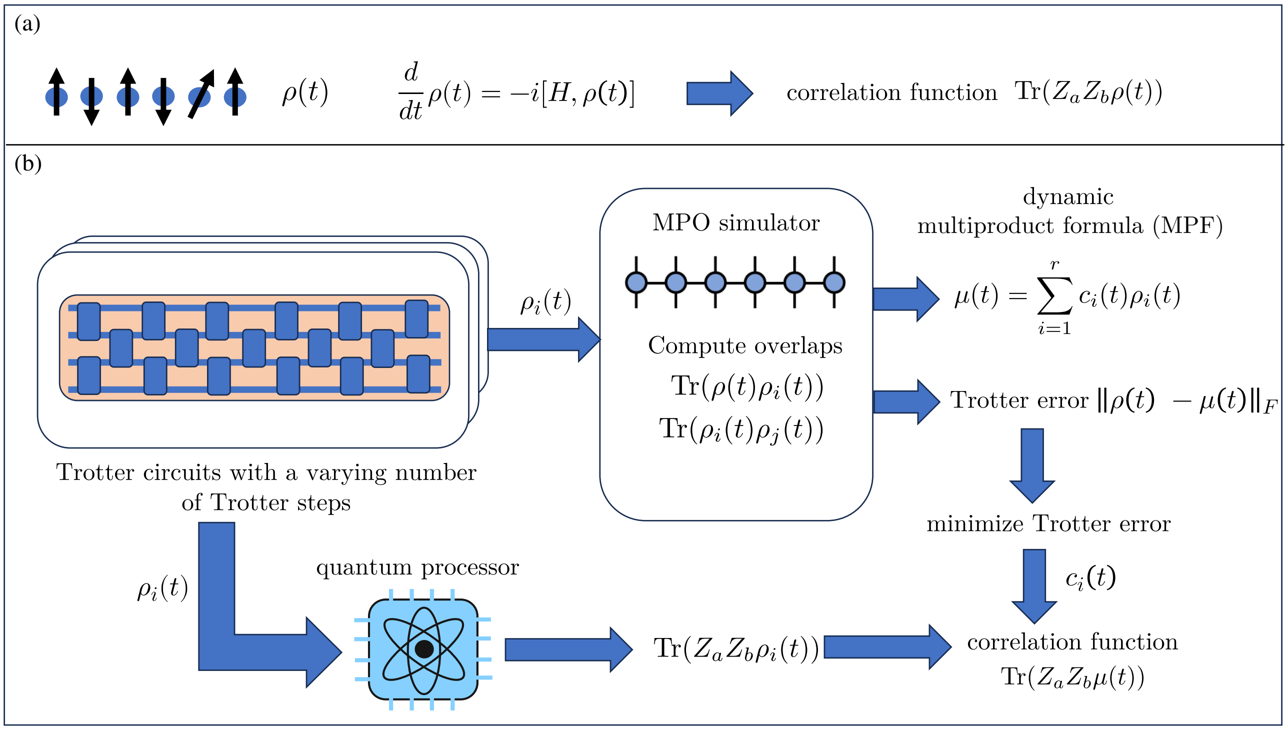

FIG. 1. (a) The Hamiltonian dynamics for a spin-chain Hamiltonian H. The goal is to compute time-dependent correlation functions such as the expected value of Pauli $\sigma_a\sigma_b$ on the time-evolved state $\rho(t)$. (b) The MPO-MPF simulator workflow. The time-evolved state $\rho(t)$ is approximated by a linear combination of states $\rho_i(t)$ generated by an ensemble of Trotter circuits with a varying number of Trotter steps. A classical simulator based on matrix product operators (MPOs) is responsible for computing overlaps for the chosen ensemble of states. Our approximation for $\rho(t)$ is a dynamic multiproduct formula (MPF) $\mu(t) = \sum_{i=1}^{r} c_i(t)\rho_i(t)$, with coefficients $c_i(t)$ optimized to minimize the Trotter error measured by the Frobenius-norm distance between $\rho(t)$ and $\mu(t)$. This distance is a simple function of the overlaps $\mathrm{Tr}(\rho(t)\rho_i(t))$ and $\mathrm{Tr}(\rho_i(t)\rho_j(t))$. A quantum processor is responsible for estimating correlation functions associated with each individual state $\rho_i(t)$. Finally, different terms in the MPF are combined to obtain an estimate $\mathrm{Tr}(\rho(t)\sigma_a\sigma_b) \approx \sum_{i=1}^{r} c_i(t)\mathrm{Tr}(\rho_i(t)\sigma_a\sigma_b)$. The classical cost of computing overlaps such as $\mathrm{Tr}(\rho(t)\rho_i(t))$ is small due to an approximate cancellation between forward and backward time evolution generated by H and by the Trotter circuit.

concretely, we calculate the MPOs corresponding to the unitaries $S(t/k_i)^{-k_i}e^{-iHt}$ and $S(t/k_i)^{-k_i}S(t/k_i)^{k_j}$ and contract them with the initial state $|\psi_0\rangle$ to obtain the fidelities $|\langle\psi_i|\psi\rangle|^2$ and $|\langle\psi_i|\psi_j\rangle|^2$.

Calculating these quantities with MPOs still incurs a truncation error due to the finite bond dimension, which is then propagated to the multiproduct coefficients c_j—this raises the immediate question of whether or not the reduction in Trotter error obtained via the implementation of an *ideal* MPF will be negated by the error incurred by the calculation of the coefficients with tensor networks with a truncated bond dimension. Of course, this error can—in principle—be reduced arbitrarily by increasing the bond dimension, but in this scenario one must then ask if the required bond dimension to do so is so large that one is in effect just simulating the quantum system classically. A significant portion of this work is dedicated to addressing these subtle questions. We prove that despite the propagation of truncation errors just outlined, truncation error does not negate Trotter error and hence our MPO-MPF algorithm is indeed advantageous as compared to either purely classical simulations based on tensor networks or purely quantum simulations based on Trotterization. In particular, we prove in Lemma B1 that there is no first-order error in the observables computed with our tensor network enhanced dynamic MPF algorithm (see Fig. 2 and points (1) and (2) in Sec. III B as well as detailed proofs and numerical demonstrations in Appendix A).

This work is structured as follows. In Sec. II, we recall the basic definitions used in previous work on MPFs. In Sec. III, we introduce our new MPO-MPF algorithm (for a summary, see Fig. 1). We discuss the errors involved and show in Fig. 2 how MPO-MPF provides an advantage over purely classical or purely quantum approaches in isolation—we leave a more detailed error analysis to Appendix A. In Sec. IV, we present the results of both our classical and quantum simulations (see Figs. 3, 4, and 9). We compare our MPO-MPF algorithm to the previously proposed static MPF algorithms [16–18] and demonstrate that MPO-MPF outperforms static MPF even in the presence of errors due to the finite truncation of the bond dimension in the tensor network. We also introduce a classically implementable numerical test to determine the simulation times at which any given MPF will provide an advantage over a single-product formula (see the inequality in Eq. (16)). We stress that this numerical test applies both to the dynamic MPF algorithm proposed here as well as to static MPFs. We conclude in Sec. V, where we discuss our MPO-MPF algorithm in the context of other quantum simulation algorithms that also aim to reduce Trotter error with short-depth circuits [21,22]. We discuss how each of these cases can be applied as part of the MPO-MPF workflow. In particular, in Appendix C, we present results of the combination of MPO-MPF and the *AQCtensor* method from Ref. [22].

II. BACKGROUND

A. Multiproduct formulas

In what follows, we briefly recall from Ref. [20] the notions of Trotter product and MPFs for quantum time evolution. A product formula is a quantum circuit $S(t)$ approximating the evolution operator e^{-itH} of a quantum system with a Hamiltonian H: a common approach is to split the time window $(0, t)$ into k intervals of length t/k, construct a circuit $S(t/k)$, and apply it k times. The depth of the resulting circuit $S(t/k)^k$ scales linearly with k, and k depends on the structure of H, the desired approximation error, and the type of product formula [20].

We stress that Trotter product formulas that accurately approximate Hamiltonian dynamics might be very deep. MPFs [16] allow for the reduction of the depth of a circuit without increasing its approximation error. The latter point can be illustrated by the following example. For a given sequence of integers $k_1, \ldots, k_r$, consider a *static MPF* in the form of a real linear combination $\mu^S(t) = \sum_{j=1}^r c_j \rho_{k_j}(t)$ of r density matrices $\rho_{k_j}(t) = S(t/k_j)^{k_j} \rho_0 S(t/k_j)^{-k_j}$, each of which approximates the exact time-evolved density matrix $\rho(t) = e^{-itH} \rho_0 e^{itH}$, where $\rho_0 = |\psi_0\rangle\langle\psi_0|$ and S is a Trotter product formula of order p. The coefficients c_j are chosen to solve the following linear system:

$$\sum_{i=1}^{r} c_i = 1 \quad \text{and} \quad \sum_{i=1}^{r} \frac{c_i}{k_i^q} = 0 \tag{1}$$

for $q \in \{p, p+1, \ldots, 2p-1\}$. This choice of coefficients guarantees that the resulting MPF μ^S has at least quadratically (or even exponentially) better accuracy (measured in 1-norm, $\|\rho - \mu^S\|_1$) than each ρ_{k_j} without using circuits that are any deeper than those used to produce ρ_{k_j} [20]. As a result, the expected value of any observable $\mathrm{Tr}(\mathcal{O}\rho(t))$ can be approximated better by a linear combination $\sum_{j=1}^r c_j \, \mathrm{Tr}(\mathcal{O}\rho_{k_j}(t))$.

By design, the static MPF $\mu^S(t)$ reduces the Trotter error: $\|\rho - \mu^S\|_1 < \|\rho - \rho_{k_j}\|_1$. However, this error is not minimized: i.e., $\min_{c_i} \|\rho - \sum_{j=1}^r c_j \rho_{k_j}\|_1 < \|\rho - \mu^S\|_1$. In addition, the coefficients c_i found from Eq. (1) are sensitive to the choice of k_i, as for certain k_i the set of linear equations in Eq. (1) is ill conditioned numerically. In this scenario, the resulting vector of coefficients has a large condition number $\sum_i |c_i|$ which leads to an amplification of the sampling noise for expectation values evaluated on a quantum computer. These two points have been addressed in Ref. [20] by introducing dynamic MPFs $\mu^D(t) = \sum_{j=1}^r c_j(t) \rho_{k_j}(t)$, with time-dependent coefficients $c_j(t)$ chosen to minimize the error $\|\mu^D(t) - \rho(t)\|_F$ measured in the Frobenius norm. It can be computed by solving a convex optimization problem in which a cost function E_F^D is minimized with respect to coefficients $c_j(t)$:

$$E_F^D = \|\rho(t) - \mu^D(t)\|_F^2 = 1 + \sum_{i,j=1}^{r} M_{ij}(t) c_i(t) c_j(t) - 2 \sum_{i=1}^{r} L_i(t) c_i(t), \tag{2}$$

where $M(t)$ is the Gram matrix, with elements

$$M_{ij}(t) = \mathrm{Tr}(\rho_{k_i}(t)\rho_{k_j}(t)) = \left| \langle\psi_0| S(t/k_i)^{-k_i} S(t/k_j)^{k_j} |\psi_0\rangle \right|^2 \tag{3}$$

and $L(t)$ is a vector of overlaps with components

$$L_j(t) = \mathrm{Tr}(\rho(t)\rho_{k_j}(t)) = \left| \langle\psi_0| S(t/k_j)^{-k_j} e^{-iHt} |\psi_0\rangle \right|^2 . \tag{4}$$

We discuss in detail the relationship between the 1-norm and the Frobenius norm in Sec. IV. In particular, we discuss the conditions under which dynamic MPF outperforms each Trotter product formula as measured in 1-norm error (see Eq. (16) and the surrounding text). It has been noted in Ref. [20, Sec. 5.a] that dynamic MPF outperforms the static one at long simulation times and it does not require that the k_i be constrained in order to get well-conditioned coefficients.

III. MPO-BASED DYNAMIC MPFs

Here, we present a MPO-based method to calculate the dynamic coefficients $c_j(t)$ classically, leaving the calculation of the expectation values $\mathrm{Tr}(\mathcal{O}\rho_{k_j})$ to the quantum computer. We argue that in the context of dynamic MPF, the bond dimension of the MPO-based method scales (in time) more favorably than that of MPSs and we also show that the error (in computing expectation values) due to the MPO bond-dimension truncation can be made negligible compared to the error of the individual Trotter product formulas used in the multiproduct.

Recall from Eqs. (3)–(4) that M_{ij} is the overlap of the quantum states $|\psi_j\rangle = S(t/k_j)^{k_j}|\psi_0\rangle$ and $|\psi_i\rangle = S(t/k_i)^{k_i}|\psi_0\rangle$ and L_j is the overlap of the quantum states $|\psi_j\rangle$ and $|\psi_{\rm ex}\rangle = e^{-iHt}|\psi_0\rangle$. For large times and large systems, it is not classically efficient to store $|\psi_i\rangle$, $|\psi_j\rangle$, and $|\psi_{\rm ex}\rangle$ as MPSs, even for 1D systems. To calculate M_{ij} and L_j and hence $c_j(t)$, we must therefore avoid explicitly storing the quantum states. To do so, we define the objects F_{ij} and $F_{{\rm ex},j}$:

$$\begin{aligned} F_{ij} &\equiv S\left(\frac{t}{k_i}\right)^{-k_i} S\left(\frac{t}{k_j}\right)^{k_j}, \\ F_{{\rm ex},j} &\equiv e^{iHt} S\left(\frac{t}{k_j}\right)^{k_j}. \end{aligned} \tag{5}$$

The quantities M_{ij} and L_j are thus given by

$$M_{ij} = |\langle\psi_0| F_{ij} |\psi_0\rangle|^2 \tag{6}$$

and

$$L_j = |\langle\psi_0| F_{{\rm ex},j} |\psi_0\rangle|^2. \tag{7}$$

In practice, we approximate $e^{-iHt} \approx S(t/k_0)^{k_0}$ with $k_0 \gg k_i$. In what follows, we thus consider $F_{{\rm ex},j}$ to be a special case of F_{ij} with $i = 0$. In Algorithm 1, we present our algorithm to calculate F_{ij}. We start by initializing F_{ij} to be the MPO representing the identity operator. We then update it by a series of left or right matrix multiplications by $S^\dagger(t/k_i)$ or $S(t/k_j)$, respectively, which can be seen as a series of backward or forward time evolutions. The goal is to keep F_{ij} as close to the identity as possible throughout the loop to minimize the bond dimension required to store the MPO.

ALGORITHM 1. MPO-based algorithm to calculate M_{ij} and L_j.

```
$F_{ij} \leftarrow \mathbb{1}$
$t_i \leftarrow 0$
$t_j \leftarrow 0$
while $t_i < t$ and $t_j < t$ do
    if $t_j \leq t_i$ then
        $F_{ij} \leftarrow F_{ij} S\left(\frac{t}{k_j}\right)$
        $t_j = t_j + \frac{t}{k_j}$
    else
        $F_{ij} \leftarrow S^\dagger\left(\frac{t}{k_i}\right) F_{ij}$
        $t_i = t_i + \frac{t}{k_i}$
    end if
end while
```

A. Memory efficiency

We now briefly comment on the memory requirements in Algorithm 1, while a more detailed analysis can be found in Appendix A. More specifically, we compare the bond dimension of the MPO required to store F_{ij} with a given precision versus the bond dimension of the MPS required to store the time-evolved state $e^{-iHt}|\psi_0\rangle$. For a general Hamiltonian (i.e., one that has not been handpicked to exhibit behavior such as many-body localization), the bond dimension required to store $e^{-iHt}|\psi_0\rangle$ scales exponentially in time, i.e., $\log(\chi_{\rm mps}) \propto T$. However, unlike the MPS for the quantum state, the bond dimension required to store F_{ij} and $F_{{\rm ex},j}$ as MPOs decreases with decreasing Trotter step dt. In Appendix A, we argue that $\log(\chi_{mpo}) \propto Tdt^2$ in the worst-case scenario. For any given simulation time t, we thus argue that there exists a Trotter step dt such that it is more efficient to store the MPOs F_{ij} than to store the MPS for the quantum states individually.

B. Error analysis

Now consider the effect of truncating the bond dimension of the MPOs (for computing overlaps M_{ij} and L_j) on the accuracy of estimating expectation values. In Appendix A, we argue that for any given truncation threshold λ_0 on the bond dimension, there exists a value of the Trotter step number k such that:

(1) The memory resources required to store F_{ij} are lower than those required to store the quantum state $e^{-iHt}|\psi_0\rangle$ using MPS.
(2) For an observable $\mathcal{O}$, MPO truncation error $\varepsilon(\lambda_0)$, Trotter error $\mathcal{E}$, and coefficients c_j obtained using Algorithm 1, one has $|{\rm Tr}(\mathcal{O}\rho(t)) - \sum_{j=1}^{r} c_j(t)\,{\rm Tr}(\mathcal{O}\rho_{k_j}(t))| \leq |{\rm Tr}(\mathcal{O}(\rho(t) - \mu^D(t)))| + O(\varepsilon(\lambda_0)\mathcal{E})$.

Point (2) suggests that the errors on the observables arising from the finite bond dimension are dominated by $|{\rm Tr}(\mathcal{O}(\rho(t) - \mu^D(t)))|$—the error of the dynamic MPF with exact M_{ij} and L_j, while the error due to MPO truncation, $O(\varepsilon(\lambda_0)\mathcal{E})$, is proportional to the product of the Trotter error and the truncation error, i.e., there are no first-order error terms arising from the tensor-network part of our algorithm (see Eq. (B2) and the surrounding text).

Below, we further illustrate point (1) above numerically for the well-studied Heisenberg model, with Hamiltonian H given by

$$H = -\sum_{i=1}^{L-1}(S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + S_i^z S_{i+1}^z), \tag{8}$$

with $S_i^x = \frac{1}{2}\sigma_i^x$, $S_i^y = \frac{1}{2}\sigma_i^y$ and $S_i^z = \frac{1}{2}\sigma_i^z$, where σ^x, σ^y, and σ^z are the Pauli matrices. We consider the initial state to be the Néel state: $|\psi_0\rangle = |1010\ldots\rangle$. In Fig. 2, we compare three different simulation methods:

(1) $\rho_{k_{\max}} = |\psi_{k_{\max}}\rangle\langle\psi_{k_{\max}}|$: $k_{\max} = \max\{k_1, k_2\}$, $|\psi_{k_{\max}}\rangle$ is the second-order Trotter product formula implemented using MPS-based classical simulation [29] with large bond dimension ($\chi = 400$).

(2) $\rho_{\mathrm{dynMPF}} = c_1|\psi_{k_1}\rangle\langle\psi_{k_1}| + c_2|\psi_{k_2}\rangle\langle\psi_{k_2}|$: c_1, c_2 computed by minimizing Eq. (2) provided that the overlaps M_{ij} and L_j are computed as per Algorithm 1 with low MPO bond dimension ($\chi = 50$), $|\psi_{k_1}\rangle$, is computed similarly to $|\psi_{k_{\max}}\rangle$.

(3) $\rho_{\mathrm{MPS}} = |\psi_{\mathrm{MPS}}(t)\rangle\langle\psi_{\mathrm{MPS}}(t)|$: $|\psi_{\mathrm{MPS}}(t)\rangle$ calculated using MPS-based classical simulation [29] with relatively low MPS bond dimension ($\chi = 100$) and a very low time step dt such that there is effectively no Trotterization error.

Each of the above methods is then characterized by its error, defined as the distance (in the Frobenious norm) to a "quasi-exact" density matrix ρ. More specifically, the error of $\rho_{k_{\max}}$ is given by

$$\begin{aligned} E_F^{k_{\max}} &= \|\rho - \rho_{k_{\max}}\|_F^2 \\ &= 2 - 2\,\mathrm{Tr}(\rho\rho_{k_{\max}}) \\ &= 2 - 2|\langle\psi_0|\,S^{k_{\max}}\left(\frac{t}{k_{\max}}\right)e^{-iHt}\,|\psi_0\rangle|^2. \end{aligned} \tag{9}$$

Here, $e^{-iHt}|\psi_0\rangle$ is computed with the fourth-order Trotter product formula, a very low time step dt, and a large bond dimension ($\chi = 400$). Similarly, the Frobenius-norm error of the MPS simulation is given by

$$E_F^{\mathrm{MPS}} = \|\rho - \rho_{\mathrm{MPS}}\|_F^2 = 2 - 2|\langle\psi_{\mathrm{MPS}}(t)|\,e^{-iHt}\,|\psi_0\rangle|^2, \tag{10}$$

where $|\psi_{\mathrm{MPS}}(t)\rangle$ is an approximation to the exact state after truncation of the bond dimension and renormalization of the singular values such that $|\langle\psi_{\mathrm{MPS}}(t)|\psi_{\mathrm{MPS}}(t)\rangle|^2 = 1$. The Frobenius norm of the dynamic MPF error is defined in Eq. (2)—however, for a fair comparison with E_F^{MPS} and $E_F^{k_{\max}}$, one must take great care with how Eq. (2) is applied. In particular, we first calculate M_{ij} and L_j using Algorithm 1 and denote the resulting quantities by M_{ij}^{MPO} and L_j^{MPO}. We then input these quantities to the right-hand side of Eq. (2) and find the coefficients c_i^{MPO} that minimize this quadratic function, subject to the constraint $\sum_i c_i = 1$. We then calculate M_{ij} and L_j quasi-exactly, i.e., using heavy numerical simulations with bond dimension $\chi = 400$. We denote the resulting quantities by M_{ij}^{ex} and L_j^{ex}. We then define the quantity $E_F^{\mathrm{MPO-MPF}}$:

$$\begin{aligned} E_F^{\mathrm{MPO-MPF}} &= 1 + \sum_{i,j=1}^{r} M_{ij}^{\mathrm{ex}}(t)c_i^{\mathrm{MPO}}(t)c_j^{\mathrm{MPO}}(t) \\ &\quad - 2\sum_{i=1}^{r} L_i^{\mathrm{ex}}(t)c_i^{\mathrm{MPO}}(t). \end{aligned} \tag{11}$$

In Fig. 2, we see that $E_F^{\mathrm{MPO-MPF}}$ has the lowest error of the three methods, despite having the lowest bond dimension in its classical part of the workflow (i.e., for the calculation of the coefficients $c_i^{\mathrm{MPO}}(t)$).

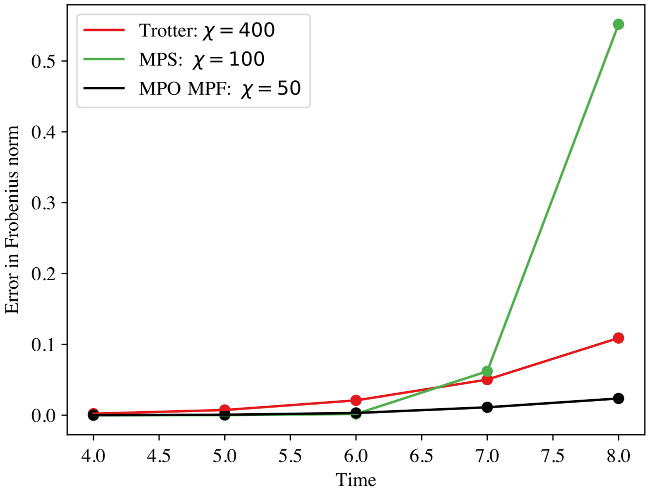

FIG. 2. The Frobenius norm of the error in the density matrices produced by three different simulation methods of the time evolution with the 50-qubit Hamiltonian in Eq. (8) acting on the initial state $|\psi_0\rangle = |1010\ldots\rangle$. Red line: $E_F^{k_{\max}}$ in Eq. (9) with $k_{\max} = k_1 = 18$, $\chi = 400$. Green line: E_F^{MPS} in Eq. (10) with $\chi = 100$ and very small time step ($dt = 0.025$). Black line: $E_F^{\mathrm{MPO-MPF}}$ in Eq. (11) with $\chi = 50$ and two Trotter formulas used in the multiproduct, $k_1 = 18$ and $k_2 = 14$. The error arising from dynamic MPO-MPF with bond dimension $\chi = 50$ is lower than pure Trotter with bond dimension $\chi = 400$ (i.e., a proxy for a quantum device) and for a pure-MPS-state simulation with bond dimension $\chi = 100$. The total memory requirements to store the MPS on L qubits, i.e., the total number of floating-point complex numbers, is given by $2L\chi_{\mathrm{mps}}^2$, while for the MPO this number is $4L\chi_{\mathrm{mpo}}^2$. The total classical memory requirements for the $\chi = 100$ MPS curve are thus larger than those of the $\chi = 50$ MPO curve. When implemented on a real quantum device, the shot noise is magnified by a factor given by the 1-norm of the coefficients $\sum_i |c_i(t)|$. One must thus ensure that a given MPF is well conditioned (see, e.g., Ref. [18]). The MPF considered here is indeed well conditioned—the 1-norm $\sum_i |c_i(t)|$ produced from the MPF algorithm is less than 4.5 for all times considered.

IV. RESULTS

In this section, we provide a thorough numerical evaluation of the quantum or classical workflow outlined in

Sec. III. In particular, we calculate the dynamic MPF coefficients $c_j(t)$ classically using Algorithm 1 and we use a quantum computer to calculate expectation values $\mathrm{Tr}(\mathcal{O}\rho_{k_j})$ for a local observable $\mathcal{O}$, namely, for one- and two-site operators σ_i^z and $\sigma_i^z\sigma_{i+1}^z$, and for a Hamiltonian similar to Eq. (8) but with adjusted coefficients to ensure that it is not integrable:

$$H = -\sum_{i=1}^{L-1}(J_i(S_i^x S_{i+1}^x + S_i^y S_{i+1}^y) + \Delta_i S_i^z S_{i+1}^z). \qquad (12)$$

In Eq. (12), the J_i are sampled from a uniform distribution supported on $[\frac{1}{4}, \frac{3}{4}]$ and $\Delta_i = 2J_i$. In what follows, we show that dynamic MPF with "shallow product formulas" can provide a comparable precision to that of a deeper (and hence more accurate) product formula in the case of classical and, more importantly, quantum simulations, thus allowing us to operate shallow circuits without loss of precision.

A. Evaluation on classical simulations

Our full workflow is as follows. First, we determine the number of Trotter steps k that are required to keep the Trotter error below a given target precision for a second-order Trotter formula, $S_2(t)$. In principle, one can obtain an estimate for the number of Trotter steps required by using the rigorous upper bound on the error [9]; however, this upper bound accounts for a worst-case scenario and so it is very likely to significantly overestimate the required number of Trotter steps in practice. Instead, we pick a large k and use Algorithm 1 to generate $F_{\mathrm{ex},k}$ in Eq. (5) and then use Eq. (9) to compute an estimate of the Trotter error using

$$E_F^k = 2 - 2|\langle\psi_0| F_{\mathrm{ex},k} |\psi_0\rangle|^2. \qquad (13)$$

Then, we pick a set of Trotter steps k_j such that $k_j < k$ for all j and we use Eq. (2) to obtain an estimate of the dynamic MPF error E_F^D arising from these k_j. The quantities M_{ij} and L_i in Eq. (2) are obtained by calculating F_{ij} and $F_{\mathrm{ex},j}$ using Algorithm 1, followed by the application of Eqs. (6) and (7). We choose the minimum values of k_j such that the resulting dynamic MPF, $\mu^D(t) = \sum_{j=1}^r c_j(t)\rho_{k_j}(t)$ of r density matrices $\rho_{k_j}(t) = S_2(t/k_j)^{k_j}\rho_0 S_2(t/k_j)^{-k_j}$, has a smaller algorithmic error than the deeper single Trotter circuit, i.e., $E_F^D(k_j) \leq E_F^k$ with $k > k_j$ for all j. In Fig. 3(a), we plot the quantity E_F^D—i.e., the error in the Frobenius norm for μ^D with three terms, $k_1 = 2$, $k_2 = 3$, and $k_3 = 4$. We compare this with the Frobenius-norm error E_F^k for a single Trotter circuit with $k = 6$ and also with a single Trotter circuit with $k = 4$. We find that the dynamic MPF with three relatively shallow circuits has a comparable algorithmic error to the deeper Trotter circuit up until $t \approx 4.1$—this crossover point is represented by the vertical dotted blue line labeled "Trotter test." We plot the same quantities in Fig. 3(b), but where E_F^D and E_F^k have been calculated by explicitly storing the states $e^{-iHt}|\psi_0\rangle$ and $S\left(t/k_j\right)^{k_j}|\psi_0\rangle$ as MPSs and calculating their overlaps. Clearly this is not a scalable approach, as the bond dimension required to do this increases exponentially with the simulation time, but we do so here for demonstration purposes and take bond dimension $\chi = 400$. We observe that the (numerically demanding) MPS simulation and the (much less numerically demanding) MPO simulation in Fig. 3 predict the same crossover point between the errors of the $k = 6$ Trotter circuit and the dynamic MPF circuits, demonstrating the resilience of the "Trotter test" to the truncation error induced by the low bond dimension used in the MPO algorithm. Furthermore, this result demonstrates the scalability of Algorithm 1 to larger timescales at which it would no longer be possible to store the quantum states accurately as MPSs.

We have so far compared the errors in the Frobenius norm of the density matrices corresponding to one Trotter product formula versus MPF but we should also consider errors measured in 1-norms, i.e., $\|\rho - \mu^D\|_1$ and $\|\rho - \rho_k\|_1$, as these quantities more closely capture the errors on the expectation values of the observables. Indeed, $\|\rho - \mu^D\|_1 = \max_{\mathcal{O}:\|\mathcal{O}\|=1}|\mathrm{Tr}(\mathcal{O}(\rho - \mu^D))|$; i.e., 1-norm represents the worst-case error of all possible expected values $|\mathrm{Tr}(\mathcal{O}(\rho - \mu^D))|$. We have the following inequalities:

$$\|\rho - \mu^D\|_F \leq \|\rho - \mu^D\|_1 \leq \sqrt{r+1}\|\rho - \mu^D\|_F, \qquad (14)$$

where r is the rank of μ^D, i.e., the number of terms in the MPF μ^D. Similarly, we have

$$\|\rho - \rho_{k_j}\|_F \leq \|\rho - \rho_{k_j}\|_1 \leq \sqrt{2}\|\rho - \rho_{k_j}\|_F. \qquad (15)$$

We can guarantee that the 1-norm of the multiproduct error is lower than the 1-norm of the Trotter error, i.e., $\|\rho - \mu^D\|_1 \leq \|\rho - \rho_{k_j}\|_1$ if the following inequality holds:

$$\sqrt{r+1}\|\rho - \mu^D\|_F \leq \|\rho - \rho_{k_j}\|_F. \qquad (16)$$

The vertical dashed black line labeled "MPF test" in Fig. 3 marks the time beyond which the inequality in Eq. (16) no longer holds. This does not mean that we can only apply MPF in the allowed time regime where Eq. (16) holds, but that if one wants to provide a rigorous guarantee that dynamic MPF with a particular set of k values will perform better than each of the individual k_j, then one must be in this allowed time regime.

We now comment on the comparison of our dynamic MPF algorithm with static MPF [18,20]. In Fig. 3, one can see that the static MPF error grows much faster than the dynamic MPF error, despite the additional error induced in the dynamic MPF from the truncation of the bond dimension. Note that the static MPF error eventually becomes

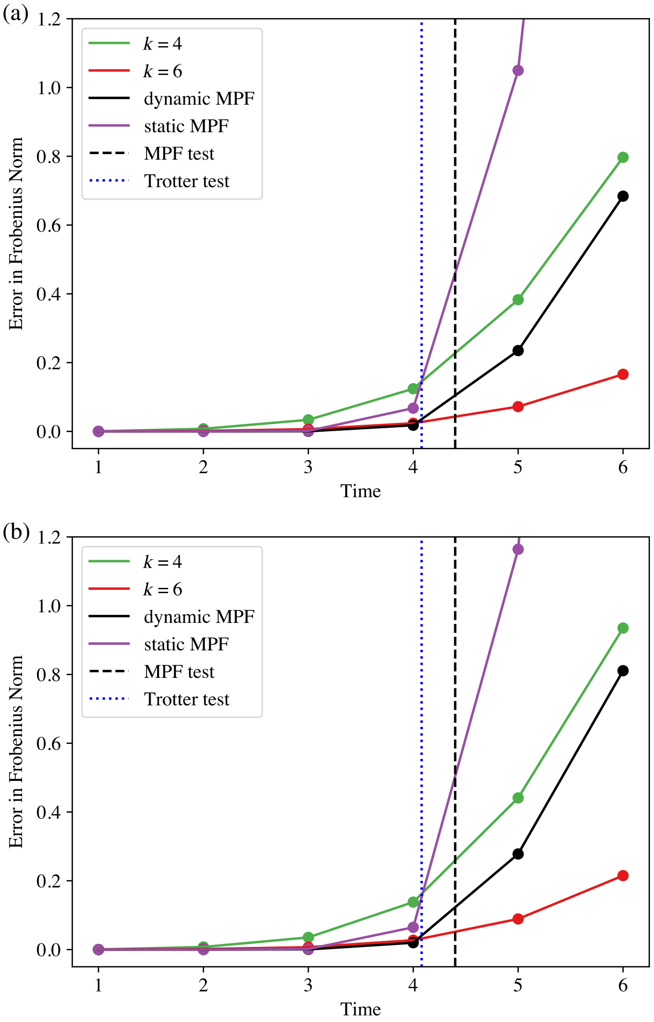

FIG. 3. Error estimates from (a) MPO and (b) MPS simulations. Green line: $E_F^{k_{\max}}$ in Eq. (9) with $k_{\max} = 4$. Red line: $E_F^{k_{\max}}$ in Eq. (9) with $k_{\max} = 6$. Black line: E_F^D in Eq. (2) with $k_1 = 2$, $k_2 = 3$ and $k_3 = 4$. Purple line: E_F^D in Eq. (2), but where the coefficients c_i are taken to be the static MPF coefficients [18,20], with $k_1 = 2$, $k_2 = 3$, and $k_3 = 4$. Vertical dotted blue line: the time at which the error on the dynamic MPF formula with $k_1 = 2$, $k_2 = 3$, $k_3 = 4$ becomes larger than the Trotter formula with $k = 6$. Vertical dashed black line: the time at which the inequality in Eq. (16) no longer holds. In (a), all quantities have been evaluated by generating the relevant MPOs using Algorithm 1, followed by the application of Eqs. (6), (7), and (13). In (b), all quantities have been calculated by explicitly storing the relevant quantum states as MPSs with $\chi = 400$ and calculating their overlaps (for discussion, see the text).

larger than that of the individual Trotter $k = 4$ Trotter circuit, which is consistent with the observations made in Ref. [20].

B. Evaluation on quantum simulations

Now, we turn to the results of the MPO-MPF algorithm on quantum hardware (see Appendix E for the details of the configuration). In Fig. 4, we plot the expectation values of two observables in the middle of the 50-qubit spin chain, namely, σ_{25}^z and $\sigma_{24}^z\sigma_{25}^z$, respectively, as measured on the quantum computer IBM_TORINO. As above, we compare a $k = 6$ Trotter circuit with a dynamic MPF with $k_1 = 2$, $k_2 = 3$, and $k_3 = 4$. These values of k_1, k_2, k_3, and k have been chosen to ensure that the algorithmic error of the dynamic multiproduct is comparable to the algorithmic error of the Trotter circuit $k = 6$; as shown in Fig. 3, this is the case up until time $t = 4.1$ for the circuits in question. By keeping the algorithmic errors roughly equal for the Trotter circuit and the dynamic multiproduct, we can accurately assess the reduction in the device error arising from the reduced circuit depth of the circuits used in the dynamic

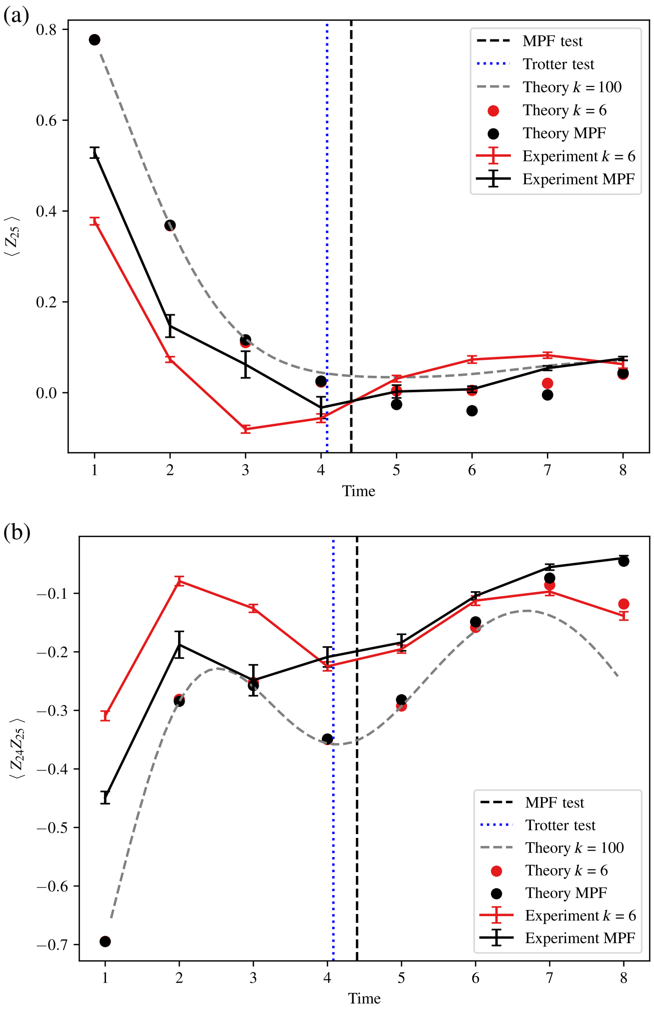

FIG. 4. The expectation values of the observables (a) σ_{25}^z and (b) $\sigma_{24}^z\sigma_{25}^z$ after time evolution with the 50-qubit Hamiltonian in Eq. (12) as measured on the quantum computer IBM_TORINO. The dynamic MPF performs better (i.e., is closer to the exact curve) than the single Trotter circuit at times before the “Trotter test” and “MPF test” (i.e., the vertical dotted blue and dashed black lines, respectively, as in Fig. 3). This is due to the reduced depths of the circuits used in the dynamic MPF.

MPF as compared to a single Trotter circuit. As shown in Fig. 4, the dynamic MPF results are more accurate than the Trotter results up to the “Trotter-test” cutoff time when implemented on quantum hardware. This is because there are fewer gates in the circuits used in the dynamic multiproduct and hence less device noise. We note that the gate noise in each circuit can be mitigated by using probabilistic error amplification (PEA). The results in Fig. 4 do *not* use PEA to mitigate these gate errors, which explains the deviations of the experimental curves from the theoretical result, but do include dynamical decoupling (DD) to suppress non-Markovian and crosstalk errors [30,31], Pauli twirling to suppress coherent errors [15], and twirled readout extinction [32] for measurement errors. For additional results with PEA implemented on IBM_KYIV, see Appendix D.

V. DISCUSSION

In this work, we have considered the question of whether one can apply state-of-the-art classical tensor network algorithms in tandem with quantum simulations to provide an advantage over applying either method in isolation. We have addressed this question by introducing a novel dynamic multiproduct algorithm based on MPOs. Unlike previous work, the dynamic coefficients $c_i(t)$ are calculated entirely *classically* and are calculated independently at each time t, thus avoiding the problem of error propagation in time. We have shown that, despite the truncation errors induced by the finite bond dimension of the MPOs used in our algorithm, this error can be made sufficiently small such that the MPO-MPF algorithm is still advantageous as compared to either a purely classical or purely quantum approach (see Fig. 2, the discussion in Sec. III B, and Appendix A). We have compared the MPO-MPF algorithm to *static* multiproduct formulas [16–20] and found that MPO-MPF outperforms static MPF despite the truncation errors induced from the truncation of the bond dimension (see Fig. 3). Furthermore, we point out that the “MPF-test” and “Trotter-test” procedures outlined in the text are applicable not just to the dynamic MPF algorithm introduced in this work but also to static MPF; thus if for any reason one wanted to apply static MPF for a particular model instead of dynamic MPF, one could, e.g., implement a workflow in which the static MPF coefficients are used to reduce Trotter error and the MPO-MPF algorithm introduced here is used to determine the time window of validity in which the static MPF should provide an advantage. This is in contrast to relying on rather conservative error bounds derived in Ref. [20].

We have demonstrated the scalability of our algorithm by implementing it on the quantum processors IBM_TORINO and IBM_KYIV (see Appendix D) and shown how our algorithm allows for the reduction in circuit depth and hence a reduction in circuit errors (see, e.g., Figs. 4 and 10). We anticipate that this work may inspire similar tensor network plus quantum algorithms for applications beyond that of the simulation of a 1D spin chain considered here; e.g., more complex models in higher dimensions. While MPS-based methods are not well suited to study models in dimensions larger than one, we note that our algorithm never explicitly stores the quantum state itself but only the MPO representing the object F defined in Eq. (5). The theoretical error analysis in Appendix A should still apply beyond 1D—more numerical work is needed to determine in what scenarios the MPO-MPF method would work well in practice in 2D or in what scenarios it would be advantageous to use a tensor network approach that goes beyond MPOs. In the latter case, the same workflow presented in this work would be applied, i.e., the quantities M_{ij} and L_j would be calculated using the tensor network and the expectation values calculated on a quantum computer. The analysis of the propagation of errors from the tensor network to the MPF presented in this work would need to be extended. We point out that any of the potential difficulties that may arise in a dynamic MPF workflow that uses 2D tensor networks will surely *also* arise in a direct classical simulation of the full quantum system that uses the same tensor network structure. However, several modern 2D tensor network methods to simulate the dynamics of 2D quantum systems have been proposed [33–36]—we therefore conjecture that many of these methods should *also* be applicable to the tensor network enhanced dynamic MPF workflow introduced in this work.

Finally, we have discussed our tensor network enhanced dynamic MPF algorithm in the context of other quantum simulation algorithms that have been designed to perform better than standard Trotter circuits [21,22]. In each of these cases, methods have been proposed to identify an optimized parametric circuit that has a lower approximation error than a Trotter circuit. This is in contrast to our dynamic MPF algorithm, which takes several different circuits that approximate the time-evolution operator and calculates the optimal projection coefficients of the ideal circuit onto the span of these approximate circuits. This projection procedure is agnostic to the form of the approximate circuits used as a basis. Thus one can always apply our tensor network based dynamic MPF algorithm in conjunction with these other approaches by taking linear combinations of the optimized circuits themselves rather than of the Trotter circuits. This is in contrast to previous works on MPFs [16–20] which are designed exclusively to take linear combinations of Trotter circuits. In Appendix C, we provide an example of MPO-MPF working in tandem with another Trotter error-minimization approach called *AQCtensor* [22]—an approximate quantum compiler based on tensor networks. This approach uses classical optimization to minimize the Trotter error of the circuit up until the latest time t that can be simulated classically.

When applied in combination with MPO-MPF, the projection coefficients $c_i(t)$ are calculated with respect to the optimized circuits found via *AQCtensor*. This demonstrates how MPO-MPF can be applied simultaneously with circuit-optimization techniques and we envisage that a similar workflow could be applied with other methods such as the one proposed in Ref. [21].

ACKNOWLEDGMENTS

We would like to thank Antonio Mezzacapo for useful discussions, guidance, and coordination of this project. We thank Nicolas Lorente, Kate Marshall, and Max Rossmannek for useful discussions.

APPENDIX A: ERROR ANALYSIS

In Sec. III, the precision and memory requirements to store F_{ij} in Algorithm 1 have been briefly discussed—we elaborate on these points here. Recall that to find the dynamic MPF coefficients $c_j(t)$, we minimize E_F^D in Eq. (2) subject to the constraint $\sum_i c_i = 1$. To do so, we must calculate the quantities M_{ij} and L_j defined in Eqs. (3) and (4), which can be rewritten in terms of F_{ij} and $F_{\mathrm{ex},j}$ (see Eq. (5)). Recall that $F_{0j} \equiv S^\dagger (t/k_0)^{k_0} S\left(tk_j/\right)^{k_j} \approx e^{iHt} S\left(t/k_j\right)^{k_j}$, with $k_0 \gg k_j$. In what follows, we thus consider the object

$$F \equiv e^{iHt} S\left(\frac{t}{k}\right)^{k} \tag{A1}$$

and we take the Hamiltonian in Eq. (12). We consider the memory requirements to generate and store F as an MPO and compare these to the requirements to store the quantum state $|\psi_t\rangle = e^{-iHt}|\psi_0\rangle$ as an MPS. In classical time-evolution algorithms such as TEBD, one stores $|\psi_t\rangle$ as an MPS by repeated applications of singular-value decomposition (SVD) and by truncating the representation to include only singular values that are larger than a target precision threshold λ_0. The bond dimension required to store the state with precision λ_0 increases exponentially with time t:

$$\chi_{\mathrm{mps}}(\lambda_0) = f(\lambda_0) e^{v_0 k dt} = f(\lambda_0) e^{v_0 t}, \tag{A2}$$

where $dt = t/k$. We demonstrate this behavior in Fig. 5. In Fig. 5(a), we plot $\log \chi_{\mathrm{mps}}$ versus t for a range of values of λ_0 and find an estimate for $f(\lambda_0)$ and v_0 by fitting a linear function to each curve. In Fig. 5(b), we plot these estimates for $f(\lambda_0)$ and v_0 versus λ_0 and find that, as expected, $f(\lambda_0)$ is strongly dependent on λ_0, whereas v_0 is approximately constant.

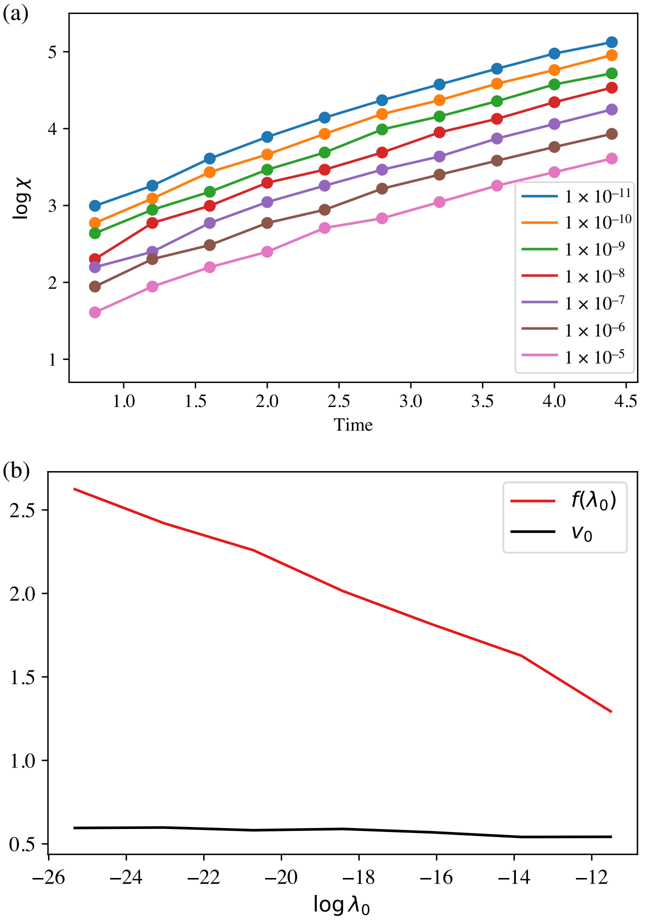

FIG. 5. (a) A demonstration of the scaling behavior of the bond dimension χ_{mps} in Eq. (A2). $\log \chi_{\mathrm{mps}}$ is plotted versus time for a range of values of the cutoff λ_0. (b) The quantities $f(\lambda_0)$ and v_0 are found by fitting a linear function to each of the curves in (a). As expected, one sees that f strongly depends on λ_0, while v_0 does not.

We expect a similar behavior for the bond dimension required to store the MPO representation of the full time-evolution operator e^{-iHt}:

$$\chi_{\mathrm{mpo}}(\lambda_0) = g(\lambda_0) e^{v_1 k dt} = g(\lambda_0) e^{v_1 t}. \tag{A3}$$

Now, instead of the unitary e^{-iHt}, we consider the bond dimension required to store the object F in Eq. (A1) with precision λ_0. We expand F using the BCH formula to obtain

$$F \approx e^{-idt^3 kC} = e^{-itdt^2 C}, \tag{A4}$$

where C is given by a sum of nested commutators. Equation (A4) suggests that one can consider F as the time-evolution operator with effective Hamiltonian $H_{\mathrm{eff}} =$

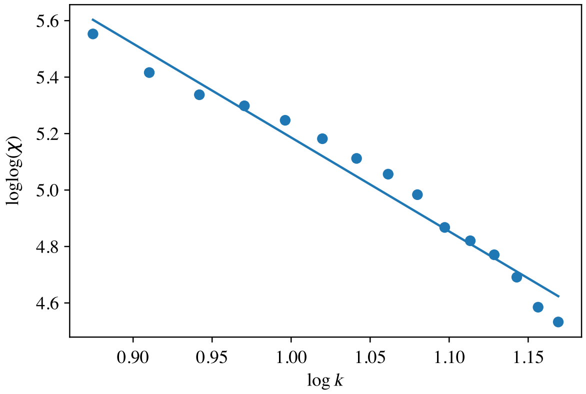

FIG. 6. A test of the scaling of χ_{mpo} with k for fixed T (see Eq. (A5)). The slope of the line is -3.7, suggesting that $\alpha \approx 4.7$ in Eq. (A5).

Cdt^2. In place of Eq. (A3), we thus write

$$\chi^F_{\text{mpo}}(\lambda_0) = g(\lambda_0)e^{v_1 k dt^\alpha} = g(\lambda_0)e^{v_1 \frac{t^\alpha}{k^{\alpha-1}}}, \tag{A5}$$

where α is some constant that should be lower bounded by 3. We test this scaling behavior in Figs. 6 and 7. In Fig. 6, we plot $\log\log(\chi^F_{mpo})$ versus $\log k$ with $t = 4$ and find that $\alpha \approx 4.7$. Similarly, in Fig. 7 we plot $\log(\chi^F_{\text{mpo}})$ versus time for fixed dt. Equations (A5) and (A2) thus show that for any desired precision λ_0 of the quantities M_{ij} and L_j in Eq. (2), there exists a value of k above which $\chi^F_{\text{mpo}}(\lambda_0) < c\chi_{\text{mps}}(\lambda_0)$ for any arbitrary constant c and hence one gains an advantage from using the MPO-MPF algorithm outlined in Sec. III over using either a purely classical MPS-based approach or a purely quantum Trotterization-based approach.

1. MPO truncation error for MPF coefficients

We now discuss the effect of the truncation error λ_0 on the MPF coefficients c_j of the dynamic MPF $\mu^D = \sum_{j=1}^r c_j \rho_{k_j}$ and on the MPF observables $\sum_{j=1}^r c_j \mathcal{O}_j$, $\mathcal{O}_j = \text{Tr}(O\rho_{k_j}(t))$.

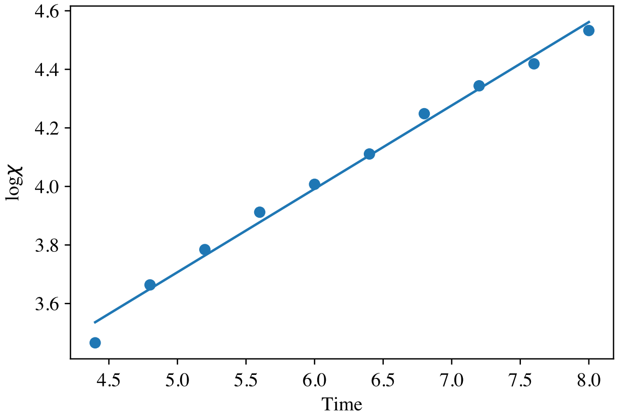

FIG. 7. A test of the scaling of χ_{mpo} with k for fixed dt.

Let $M^{(\ell_0)}$ and $L^{(\ell_0)}$ denote MPO approximations of the exact matrix M and vector L defined in Eq. (4). Let $\mathcal{E}_M^{(\ell_0)} = M^{(\lambda_0)} - M$ be the matrix of MPO-approximation errors and set $\mathcal{E}_L^{(\ell_0)} = L^{(\ell_0)} - L$. Let $\vec{c}^{\,\star}$ denote the vector of exact dynamic MPF coefficients, the unique minimizer of Eq. (2) with exact M and L, and let $\vec{c}^{\,(\ell_0)}$ denote the unique minimizer of Eq. (2) with M and L substituted by $M^{(\ell_0)}$ and $L^{(\ell_0)}$, respectively. Clearly, $\vec{c}^{\,\star}$ can be computed by solving the linear equation $M\vec{c}^{\,\star} = L + \mu\mathbb{1}$, where Lagrange multiplier μ is chosen so that $\langle \mathbb{1}|\vec{c}^{\,\star}\rangle = 1$: $\vec{c}^{\,\star} = M^{-1}L + \mu M^{-1}\mathbb{1}$ and $\mu^\star = (1 - \langle \mathbb{1}|M^{-1}|L\rangle)/\langle \mathbb{1}|M^{-1}|\mathbb{1}\rangle$.

APPENDIX B

Lemma B1. Let $\vec{c}^{\,(\ell_0)} = (M^{(\ell_0)})^{-1}(L^{(\ell_0)} + \mu\mathbb{1})$ and let μ be chosen to satisfy $\langle \mathbb{1}|\vec{c}^{\,(\ell_0)}\rangle = 1$. Then, for any $1 > \varepsilon > 0$, there exist λ_0 such that $\|L - L^{(\ell_0)}\|_2, \|M - M^{(\ell_0)}\| < \varepsilon < 1$ and

$$\|\vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}\|_2 \leq \varepsilon \frac{\|M^{-1}\| + \|\vec{c}^{\,\star}\|_2}{1 - \varepsilon}. \tag{B1}$$

For $\mathcal{O} = (\mathcal{O}_1 \dots \mathcal{O}_r)^\top$, let $\mathcal{E}(t) = O^\star\mathbb{1} - \mathcal{O}$ be the vector of Trotter errors, where $O^\star = \text{Tr}(O\rho(t))$, ρ—the exact density matrix. Then,

$$|\langle \vec{c}^{\,(\ell_0)}|\mathcal{O}\rangle - O^\star| \leq |\langle \vec{c}^{\,\star}|\mathcal{E}\rangle| + \varepsilon \frac{\|M^{-1}\| + \|\vec{c}^{\,\star}\|_2}{1 - \varepsilon}\|\mathcal{E}\|_2. \tag{B2}$$

Proof. Using the notion of the square root of a symmetric positive semidefinite matrix, namely, $M^{1/2}M^{1/2} = M$, it is easy to see that

$$\begin{aligned}
&\|\vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}\|^2 \\
&\quad = \langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|M^{1/2}M^{-1}M^{1/2}|\vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}\rangle \\
&\quad \leq \|M^{-1}\| \cdot \langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|M|\vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}\rangle.
\end{aligned} \tag{B3}$$

Let us upper bound $\langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|M|\vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}\rangle$. Since $\vec{c}^{\,\star} = M^{-1}L + \mu M^{-1}\mathbb{1}$ and $\langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|\mathbb{1}\rangle = 0$, it follows that

$$\begin{aligned}
&\langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|M|\vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}\rangle \\
&\quad = \langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|L\rangle + \mu\langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|\mathbb{1}\rangle \\
&\qquad - \langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|M|\vec{c}^{\,(\ell_0)}\rangle \\
&\quad = \langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|L\rangle - \langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|M|\vec{c}^{\,(\ell_0)}\rangle.
\end{aligned}$$

Substituting $M = M^{(\ell_0)} + M - M^{(\ell_0)}$ in $\langle \vec{c}^{\,\star} - \vec{c}^{\,(\ell_0)}|M|\vec{c}^{\,(\ell_0)}\rangle$ and recalling that $M^{(\ell_0)}\vec{c}^{\,(\ell_0)} = L^{(\ell_0)} + \mu\mathbb{1}$, with

μ chosen to satisfy $\langle\mathbb{1}|\vec{c}^{(\ell_0)}\rangle = 1$, we obtain

$$\begin{aligned}
&\langle\vec{c}^\star - \vec{c}^{(\ell_0)}|M|\vec{c}^\star - \vec{c}^{(\ell_0)}\rangle = \langle\vec{c}^\star - \vec{c}^{(\ell_0)}|L - L^{(\ell_0)}\rangle \\
&\quad - \mu\langle\vec{c}^\star - \vec{c}^{(\ell_0)}|\mathbb{1}\rangle - \langle\vec{c}^\star - \vec{c}^{(\ell_0)}|M - M^{(\ell_0)}|\vec{c}^{(\ell_0)}\rangle \\
&\quad = \langle\vec{c}^\star - \vec{c}^{(\ell_0)}|L - L^{(\ell_0)}\rangle - \langle\vec{c}^\star - \vec{c}^{(\ell_0)}|M - M^{(\ell_0)}|\vec{c}^{(\ell_0)}\rangle \\
&\quad \le \|\vec{c}^\star - \vec{c}^{(\ell_0)}\|_2(\|L - L^{(\ell_0)}\|_2 + \|M - M^{(\ell_0)}\|\|\vec{c}^{(\ell_0)}\|_2) \\
&\quad \le \|\vec{c}^\star - \vec{c}^{(\ell_0)}\|_2(\|L - L^{(\ell_0)}\|_2 \\
&\qquad + \|M - M^{(\ell_0)}\|\|\vec{c}^\star - \vec{c}^{(\ell_0)}\|_2\|\vec{c}^\star\|_2).
\end{aligned}$$

Substituting this bound into the right-hand side of Eq. (B3), dividing the result by $\|\vec{c}^\star - \vec{c}^{(\ell_0)}\|_2$, and isolating $\|\vec{c}^\star - \vec{c}^{(\ell_0)}\|_2$ on the left-hand side of the resulting inequality, we obtain Eq. (B1) provided that $\|M - M^{(\ell_0)}\| < 1$.

Now, Eq. (B2) follows by using Cauchy inequality to upper bound $|\langle\vec{c}^{(\ell_0)}|\mathcal{O}\rangle - \langle\vec{c}^{(\ell_0)}|O^\star\mathbb{1}\rangle|$,

$$\begin{aligned}
&|\langle\vec{c}^{(\ell_0)} \pm \vec{c}^\star|\mathcal{O}\rangle - \langle\vec{c}^{(\ell_0)}|O^\star\mathbb{1}\rangle| \\
&\quad \le \|\vec{c}^{(\ell_0)} - \vec{c}^\star\|_2\|\mathcal{E}\|_2 + |\langle\vec{c}^\star|\mathcal{E}\rangle|,
\end{aligned}$$

followed by Eq. (B1). This concludes the proof. ■

APPENDIX C: *AQCtensor* ALGORITHM

The *AQCtensor* algorithm has been proposed in Ref. [22] and we briefly recall it here. The algorithm aims to find a short-depth quantum circuit approximating a given MPS. It is based on classical optimization; one starts with a target quantum state $|\psi_t\rangle$ and a variational circuit ansatz $V(\Theta)|\psi_0\rangle$, with $|\psi_0\rangle$ some initial product state. Both the target state and the ansatz state are stored as MPSs. The cost function,

$$C(\Theta) = 1 - \left|\langle\psi_0|\, V^\dagger(\Theta)\, |\psi_t\rangle\right|^2, \tag{C1}$$

is evaluated by computing the overlaps of the MPS $|\psi_t\rangle$ and $V(\Theta)|\psi_0\rangle$ and is minimized with respect to rotation angles Θ, resulting in a short-depth circuit $V(\Theta)|\psi_0\rangle$ representing the target state. In this work, we employ *AQCtensor* to get a short-depth representation of the target state $|\psi_{t_1}\rangle = e^{-iHt_1}|\psi_0\rangle$. The idea is to generate $|\psi_{t_1}\rangle$ using a classical algorithm such as TEBD for the latest possible time t_1 that can be stored for a given maximum bond dimension $\chi_{\max}$ and for a given desired precision determined by the bond-dimension truncation threshold, which we denote by λ_0. Instead of using standard Trotterization to prepare the state $|\psi_{t_1}\rangle$ (which may require a deep circuit), one uses the optimized angles Θ_{opt} (obtained by minimizing the cost function) to prepare $V(\Theta_{\text{opt}})|\psi_0\rangle$. Finally, one applies a Trotter circuit to this state to obtain a time-evolved state beyond the time at which it could be stored classically: $S(t_2/k)^k V(\Theta_{opt})|\psi_0\rangle$, such that the total evolution time simulated by the full circuit is $t = t_1 + t_2$. As discussed in Ref. [22], we use a circuit ansatz $V(\Theta)$ that

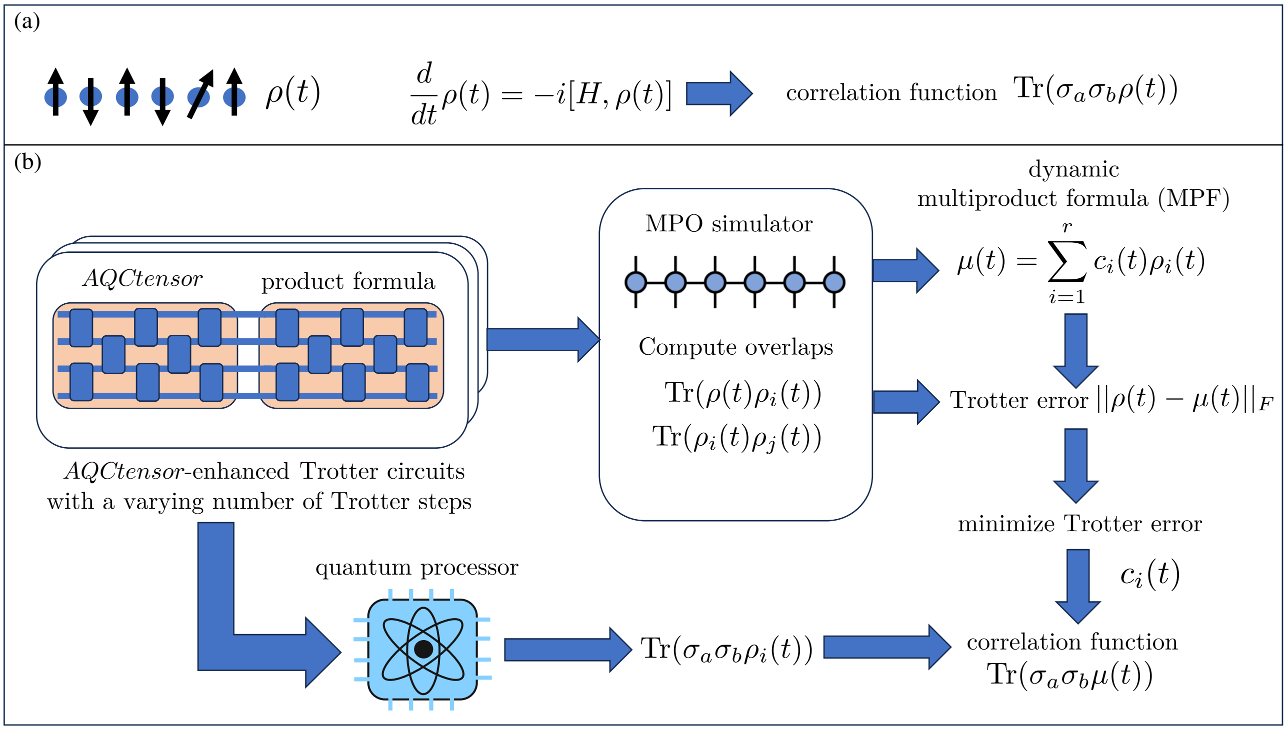

FIG. 8. The workflow from Fig. 1 but where each Trotter circuit has been compressed with the *AQCtensor* algorithm [22]: (a) the Hamiltonian dynamics for a spin-chain Hamiltonian H; (b) the MPO-MPF simulator workflow.

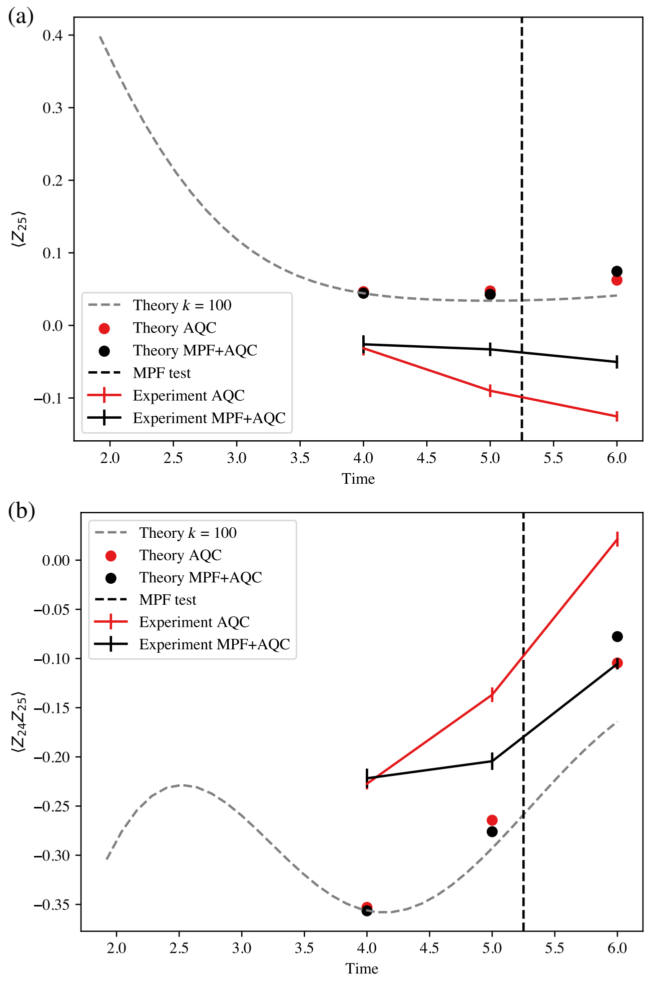

FIG. 9. The expectation values of the same observables (a) σ^z_{25} and (b) $\sigma^z_{24}\sigma^z_{25}$ as considered in Fig. 4 measured on IBM_TORINO but where the circuits have been enhanced by the *AQCtensor* algorithm. In particular, the system is simulated almost exactly using a classical MPS-based algorithm up to time $t_1 = 3$ and an optimized quantum circuit is found that closely approximates this exact evolution up to time t_1. This optimized circuit has the same depth and same structure as a Trotter circuit with $k = 2$. Additional Trotter circuits with $k = 2$ and $k = 1$ are appended to the optimized circuit to simulate times beyond $t = 3$. The red curve represents the results of the *AQCtensor* plus $k = 2$ Trotter circuit and the black curve represents the result of combining the two circuits with our dynamic multiproduct algorithm. The vertical dashed black line represents the "MPF test," which occurs at a later time than in Fig. 4 due to the effect of *AQCtensor* (see the main text).

has the same CNOT structure as a second-order Trotter circuit, allowing for the use of a "smart-initialization" scheme in which the initial values Θ_0 are set such that $V(\Theta_0)$ corresponds exactly to a second-order Trotter circuit. Therefore, after one step of a minimization algorithm such as gradient descent, it is assured that the optimized circuit ansatz will more closely represent the exact time-evolved circuit than a standard Trotter circuit.

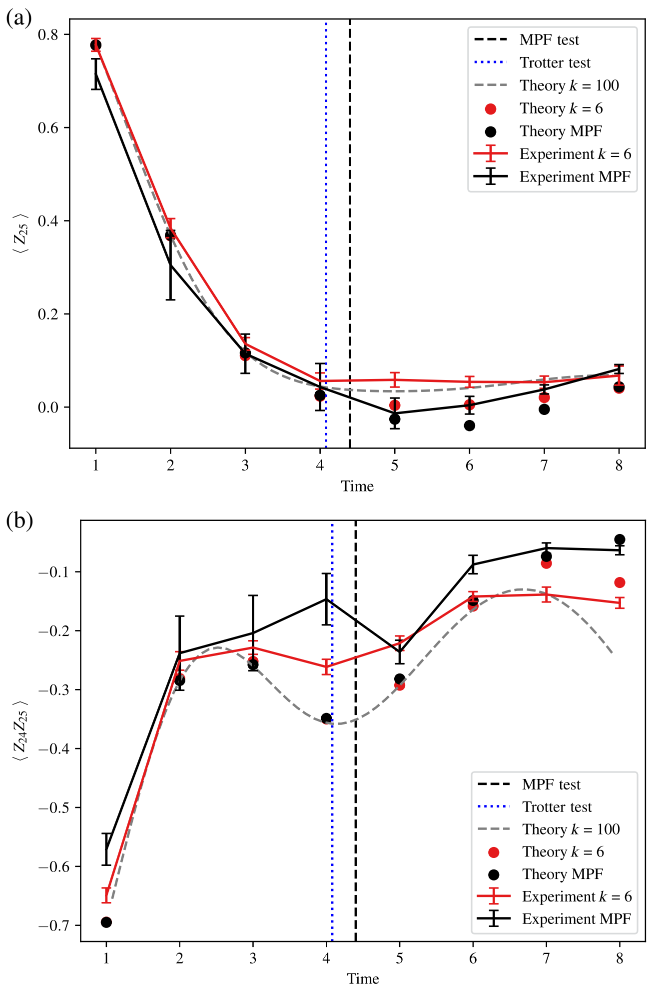

FIG. 10. The expectation values of the observables (a) σ^z_{25} and (b) $\sigma^z_{24}\sigma^z_{25}$ after time evolution with the Hamiltonian in Eq. (12) as measured on IBM_KYIV. Both MPF and $k = 6$ results are shown after incorporating probabilistic error amplification (PEA). Dynamic MPF and $k = 6$ results both give comparable results even though the $k = 6$ circuit is considerably deeper than the circuits used for the MPF.

We now consider the combination of our dynamic MPF algorithm with the *AQCtensor* algorithm (for an updated illustration of the MPO-MPF workflow when combined with *AQCtensor*, see Fig. 8). In Fig. 9, we consider the same quantities as measured on the same quantum device as in Fig. 4 but where both curves have been enhanced by the *AQCtensor* algorithm. More precisely, for each value of time t in Fig. 9, we consider two windows of time t_1 and t_2 such that $t = t_1 + t_2$. The *AQCtensor* algorithm finds a circuit of the same depth as a $k = 2$ Trotter circuit that approximates the exact time evolution e^{-iHt} up until a time t_1 quasi-exactly, i.e., with 0.99 fidelity. A standard Trotter circuit with either $k = 2$ or $k = 1$ that approximates the time-evolution operator for a time window of length t_2 is then appended to the optimized circuit to simulate times t

that are larger than t_1. In this instance, we take $t_1 = 3$ and hence $t_2 = t - 3$. In general, the value of t_1 can be taken to be the largest value for which the available classical resources can simulate the quantum system, with a given desired precision. For each value of t in Fig. 9, we have two quantum circuits, one with the t_2 time window divided into $k = 2$ Trotter steps and one where t_2 is approximated with $k = 1$ Trotter steps. The red curve represents the results of the first of these circuits and the black curve represents the result of combining the two circuits with a dynamic multiproduct. In the *AQCtensor* workflow, the observables can be calculated quasi-exactly using the classical MPS-based method for times up until $t = t_1$; hence we do not include results from quantum hardware for $t < t_1$. To calculate the dynamic multiproduct coefficients $c_j(t)$ for these *AQCtensor* circuits, we use Algorithm 1 and the method outlined in Sec. III but with a slight modification of the quantity F_{ij} in Eq. (A1). In particular, since we now divide each time t into two time windows t_1 and t_2, where for times $t < t_1$ the system is simulated quasi-exactly using the *AQCtensor* algorithm, we must define the quantities F_{ij}^{AQC} and $F_{\mathrm{ex},j}^{\mathrm{AQC}}$:

$$\begin{aligned} F_{ij}^{\mathrm{AQC}} &\equiv e^{iHt_1} S\left(\frac{t_2}{k_i}\right)^{-k_i} S\left(\frac{t_2}{k_j}\right)^{k_j} e^{-iHt_1}, \\ F_{\mathrm{ex},j}^{\mathrm{AQC}} &\equiv e^{iH(t_1+t_2)} S\left(\frac{t_2}{k_j}\right)^{k_j} e^{-iHt_1}. \end{aligned} \tag{C2}$$

The quantities M_{ij} and L_j are then calculated as described in Sec. III and the full MPO-MPF algorithm proceeds as previously described. Finally, we comment on the "MPF test" from Eq. (16) in the context of the *AQCtensor* + dynamic MPF combination. Since the application of *AQCtensor* evolves the system almost exactly for a time t_1, one can expect both the Trotter errors and MPF errors to be reduced and for the inequality in Eq. (16) to hold for longer total simulation times t. In Fig. 9 we show that this is indeed the case; the vertical line representing the final time at which the inequality in Eq. (16) holds occurs later than in Fig. 4. Thus one of the key advantages of using *AQCtensor* is that it allows for the application of our MPO-MPF algorithm at times later than would otherwise be feasible. We note that a similar workflow could instead combine the MPO-MPF algorithm with other approaches that aim to find optimized circuits for the purpose of simulating time evolution such as the one proposed in Ref. [21]. In this case, the circuits used in the dynamic multiproduct would be the optimized circuits instead of Trotterized circuits or *AQCtensor* optimized circuits and the quantities in Eq. (C2) would need to be adjusted.

APPENDIX D: PROBABILISTIC ERROR AMPLIFICATION

Before any error mitigation and noise-model simplification, the general noise channel has the Kraus operator-sum representation (KOSR)

$$\tilde{\Lambda}(\rho) = \sum_{\alpha,\beta} c_{\alpha,\beta} P_\alpha \rho P_\beta^\dagger,$$

where $P_\alpha \in \mathcal{P}_n$ is the n-qubit Pauli group.

Performing Pauli twirling over the Pauli group, which can be easily implemented as single-qubit gates, results in

$$\Lambda(\rho) = \mathbb{E}_\alpha\left[P_\alpha^\dagger \tilde{\Lambda}\left(P_\alpha^\dagger \rho P_\alpha\right) P_\alpha^\dagger\right] = \sum_\alpha c_\alpha P_\alpha \rho P_\alpha^\dagger.$$

Owing to the Pauli twirling, the Pauli transfer matrix becomes diagonal, i.e., it goes from $T_{\tilde{\Lambda}}[a,b] = (1/2^n)\,\mathrm{Tr}\left[P_a^\dagger\left(\tilde{\Lambda}\left(P_b\right)\right)\right]$ to $T_\Lambda[a,b] = \delta_{a,b} T_{\tilde{\Lambda}}[a,b]$. We supplement Pauli twirling with DD; in particular, the Carr-Purcell-Meiboom-Gill (CPMG) sequence, i.e., the $f_{\tau/2} X_p f_\tau X_m f_{\tau/2}$ sequence. Here, f_τ signifies free evolution for time τ and X_p and X_m are the $\pm\pi/2$ rotation about the X axis. The CPMG sequence cancels Z terms in the interaction Hamiltonian and, more pertinently, when applied on alternating qubits, cancels the ZZ crosstalk prevalent in superconducting qubits. DD combined with Pauli twirling suppresses coherent, crosstalk, and non-Markovian errors. In turn, this allows us to model the device noise as a sparse

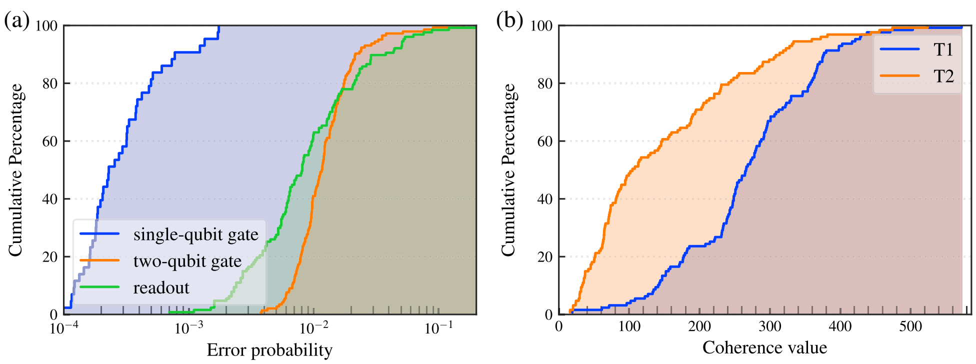

FIG. 11. The measured properties of the IBM_KYIV device. (a) The single-qubit gate error 0.04% (0.023%) mean (median), the two-qubit gate error 1.5% (1.2%) mean (median), and the readout error 1.6% (.81%) mean (median). (b) The T1 270-(270-)µs mean (median) and the T2 150-(100-)µs mean (median).

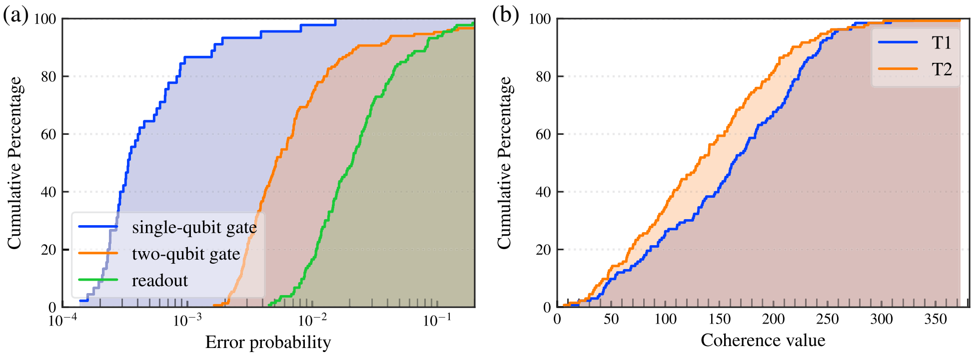

FIG. 12. The measured properties of the IBM_TORINO device. (a) The single-qubit gate error 0.11% (0.034%) mean (median), the two-qubit gate error 4.3% (0.51%) mean (median), and the readout error 3.5% (2.1%) mean (median). (b) The T1 160-(160-)μs mean (median) and the T2 130-(130-)μs mean (median).

Pauli-Lindblad model of the form

$$\Lambda(\rho) = \prod_{k\in\mathcal{K}} \left(\frac{e^{-2\lambda_k}+1}{2}\rho + \frac{e^{-2\lambda_k}-1}{2} P_k \rho P_k^{\dagger} \right).$$

Here, the $\mathcal{K}$ correspond to single and two-qubit Pauli terms on the qubits on which gates are being actively applied [37].

The first step in PEA is to learn the model coefficients λ_k. Given the circuit for Trotter evolution, we identify unique gate layers and then perform benchmark circuits for these layers at different depths to empirically determine the λ_k. Equipped with the model coefficients, we can then execute the circuit at various noise scales α. Multiplying the noise coefficients λ_k by α gives a Pauli map with the noise scaled by α, which in turn allows us to use zero noise extrapolation to find the ideal values of the desired

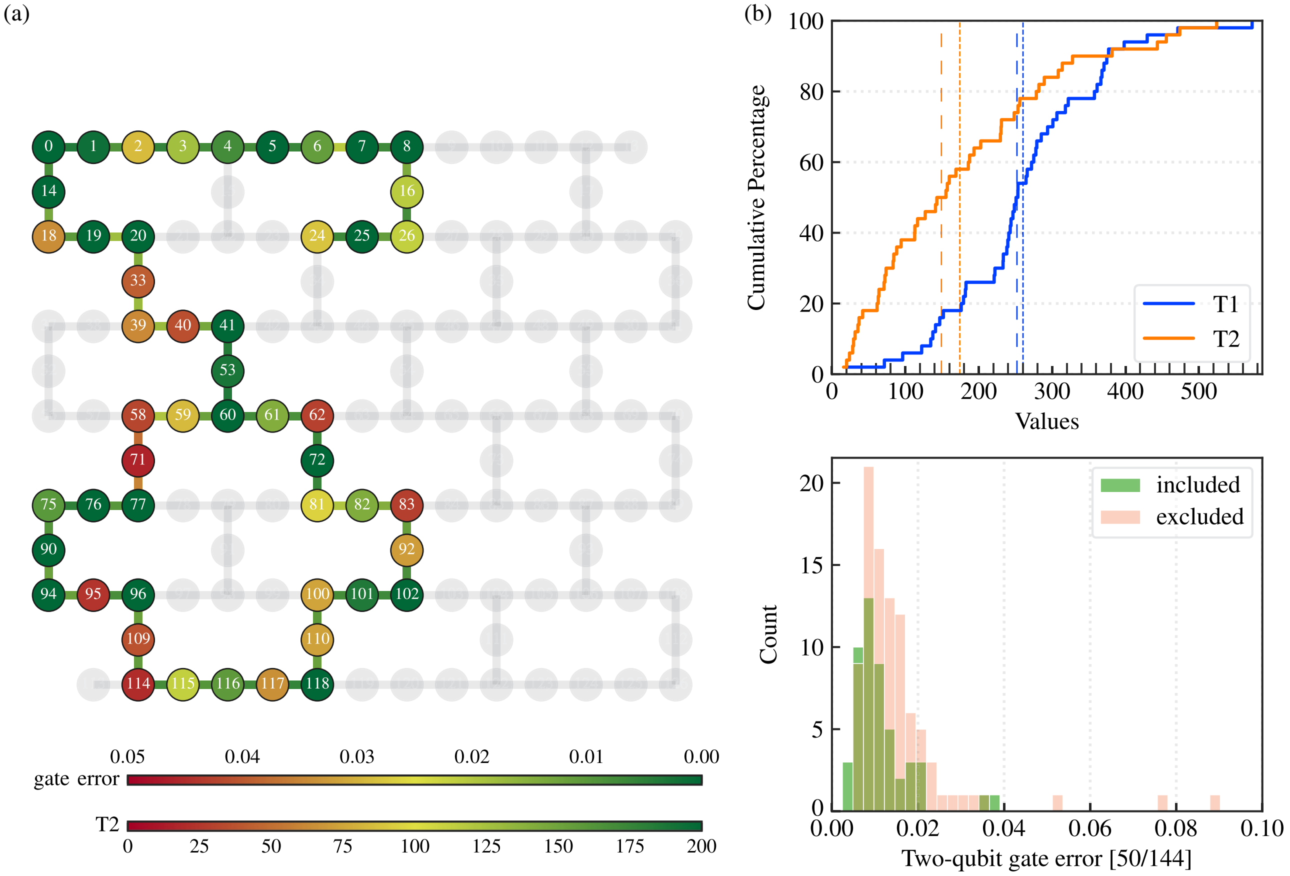

FIG. 13. The qubit subset used in the experiment on IBM_KYIV. (a) The device topology with the used qubits or edges highlighted according to their measured error rates (T2 and two-qubit gate error). (b) (top) the cumulative distribution of coherence times T1 and T2 with highlighted mean (median) values as dotted (dashed) lines and (bottom) the histogram of the two-qubit error rates for edges included in or excluded from the chosen subset.

observable. It is necessary to perform randomized twirling and DD through both the learning and the amplification steps of PEA, as these error-suppression steps enforce that noise model can be approximated using a Pauli-Lindblad model.

The results in Fig. 10 from IBM_KYIV have been acquired by combining MPF with PEA. As MPF focuses solely on addressing mitigating Trotter errors, any decoherence that causes the underlying product formula to deviate from the noiseless expectation values propagates as errors in the MPF results. Therefore, in theory, MPF is both complementary to and aided by error-suppression and error-mitigation methods that must be implemented before the product formulas are used in combination.

APPENDIX E: DEVICE SPECIFICATIONS

1. Device overview

The experimental results reported in Sec. IV have been obtained on two different devices, with different architectures. IBM_KYIV is a so-called Eagle-type quantum processor, consisting of fixed-frequency transmon qubits with capacitive coupling between neighboring qubits arranged in a heavy-hexagonal lattice of 127 qubits. IBM_TORINO is a Heron-type device, which consists of fixed-frequency transmon qubits with flux-tunable coupling, arranged in a heavy-hexagonal lattice of 133 qubits. The latter Heron-type devices are newer-generation devices compared to the Eagle ones, where the crosstalk effects due to the always-on capacitive coupling between neighboring qubits have been greatly reduced by allowing this coupling to be tunable and effectively ensuring no coupling between neighbors unless an entangling interaction between them is applied. Moreover, the tunable coupling allows for stronger (thus faster) interaction between qubits, reducing the gate time of two-qubit operations in this architecture to be the same (in terms of the order of magnitude) as single-qubit operations. This often results in the need for less complicated error-suppression and/or error-mitigation techniques for the experiments involving IBM_TORINO, as one of the main contributions to the noise in Eagle-type devices comes from the crosstalk terms arising in the

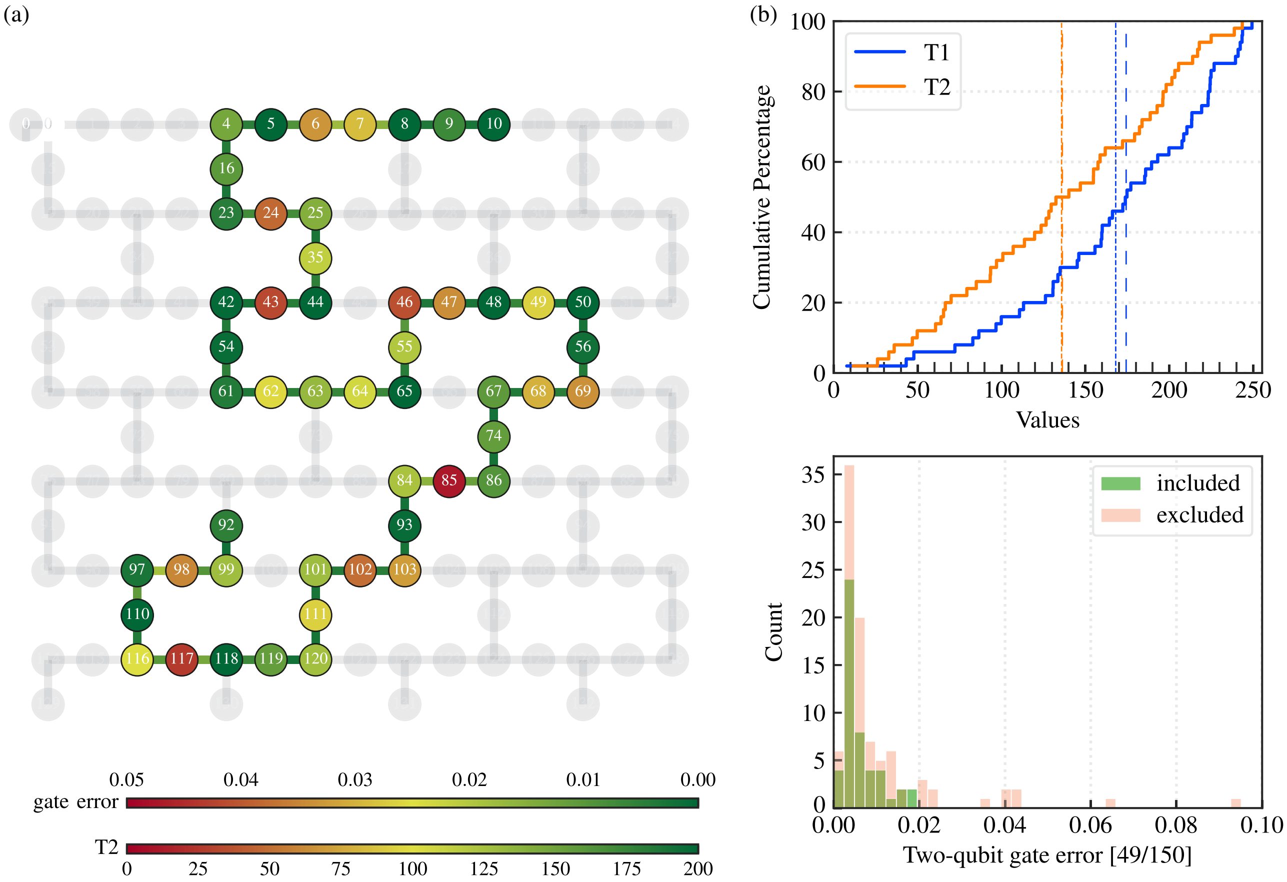

FIG. 14. The qubit subset used in the experiment on IBM_TORINO. (a) The device topology with the used qubits or edges highlighted according to their measured error rates (T2 and two-qubit gate error). (b) (top) the cumulative distribution of coherence times T1 and T2 with highlighted mean (median) values as dotted (dashed) lines and (bottom) the histogram of two-qubit error rates for edges included in or excluded from the chosen subset.

presence of always-on coupling between qubits during entangling gates.

2. Device properties

The error properties of the IBM Quantum Eagle processor are reported in Fig. 11, while those of the Heron processor are shown in Fig. 12. In Figs. 11(a) and 12(a), we show the single-qubit gate error, characterized by the randomized-benchmarking technique, the two-qubit gate error, also assessed via randomized benchmarking, and the readout error, representing the readout-assignment infidelity. In Figs. 11(b) and 12(b), we focus on coherence times, the T_1 relaxation time, the period that a qubit takes to relax to its ground state, and the T_2 dephasing time, measuring the time over which a qubit maintains its quantum state coherence.

3. Qubits used in the experiments

The qubit subsets chosen on the different devices are presented in Figs. 13 and 14 for IBM_KYIV and IBM_TORINO, respectively. The subsets have been chosen by leveraging the native capabilities of the QISKIT transpiler. This uses the VF2 graph isomorphism algorithm to find subsets of device qubits the interaction graphs of which are isomorphic to the one of the routed (i.e., where SWAP gates are added to account for device connectivity) circuit. The isomorphic layouts are then scored using a heuristic map [38] based on the error rates reported for the devices. The layout with the best (lowest) score is selected as the layout for the experiment.

[1] N. Harle, O. Shtanko, and R. Movassagh, Observing and braiding topological Majorana modes on programmable quantum simulators, Nat. Commun. **14**, 2286 (2023).

[2] J. Knolle, D. L. Kovrizhin, J. T. Chalker, and Roderich Moessner, Dynamics of a two-dimensional quantum spin liquid: Signatures of emergent majorana fermions and fluxes, Phys. Rev. Lett. **112**, 207203 (2014).

[3] J. Maldacena, S. H. Shenker, and D. Stanford, A bound on chaos, J. High Energy Phys. **2016**, 1 (2016).

[4] A. I. Larkin and Yu. N. Ovchinnikov, Quasiclassical method in the theory of superconductivity, Sov. Phys. JETP **28**, 1200 (1969).

[5] S. H. Shenker and D. Stanford, Multiple shocks, J. High Energy Phys. **2014**, 1 (2014).

[6] A. Kitaev, Hidden correlations in the Hawking radiation and thermal noise, in *Talk given at the Fundamental Physics Prize Symposium* (2014), Vol. 10, p. 33.

[7] P. Calabrese and J. Cardy, Evolution of entanglement entropy in one-dimensional systems, J. Stat. Mech.: Theory Exp. **2005**, P04010 (2005).

[8] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[9] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter error with commutator scaling, Phys. Rev. X **11**, 011020 (2021).

[10] J. Haah, M. B. Hastings, R. Kothari, and G. Hao Low, Quantum algorithm for simulating real time evolution of lattice Hamiltonians, SIAM J. Comput. **52**, FOCS18 (2021).

[11] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, Proc. Natl. Acad. Sci. **115**, 9456 (2018).

[12] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, Phys. Rev. Lett. **118**, 010501 (2017).

[13] A. M. Childs, A. Ostrander, and Y. Su, Faster quantum simulation by randomization, Quantum **3**, 182 (2019).

[14] S. Bravyi, O. Dial, J. M. Gambetta, D. Gil, and Z. Nazario, The future of quantum computing with superconducting qubits, J. Appl. Phys. **132**, 160902 (2022).

[15] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, Phys. Rev. Lett. **91**, 147902 (2003).

[16] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations. arXiv:1202.5822.

[17] G. H. Low, V. Kliuchnikov, and N. Wiebe, Well-conditioned multiproduct Hamiltonian simulation. arXiv:1907.11679.

[18] A. Carrera Vazquez, D. J. Egger, D. Ochsner, and S. Woerner, Well-conditioned multi-product formulas for hardware-friendly Hamiltonian simulation, Quantum **7**, 1067 (2023).

[19] G. Rendon, J. Watkins, and N. Wiebe, Improved error scaling for trotter simulations through extrapolation. arXiv:2212.14144.

[20] S. Zhuk, N. F. Robertson, and S. Bravyi, Trotter error bounds and dynamic multi-product formulas for Hamiltonian simulation, Phys. Rev. Res. **6**, 033309 (2024).

[21] R. Mansuroglu, F. Fischer, and M. J. Hartmann, Problem-specific classical optimization of Hamiltonian simulation, Phys. Rev. Res. **5**, 043035 (2023).

[22] N. Robertson, A. Akhriev, J. Vala, and S. Zhuk, Approximate quantum compiling for quantum simulation: A tensor network based approach, ACM Trans. Quantum Comput. **6**, 20 (2025).

[23] G. Vidal, Efficient simulation of one-dimensional quantum many-body systems, Phys. Rev. Lett. **93**, 040502 (2004).

[24] J. Haegeman, J. Ignacio Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, Time-dependent variational principle for quantum lattices, Phys. Rev. Lett. **107**, 070601 (2011).

[25] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, Unifying time evolution and optimization with matrix product states, Phys. Rev. B **94**, 165116 (2016).

[26] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, Time-evolving a matrix product state with long-ranged interactions, Phys. Rev. B **91**, 165112 (2015).

[27] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011).

[28] S. Filippov, M. Leahy, M. A. C. Rossi, and G. García-Pérez, Scalable tensor-network error mitigation for near-term quantum computing. arXiv:2307.11740.

[29] J. Hauschild and F. Pollmann, Efficient numerical simulations with tensor networks: Tensor network PYTHON (TeNPy). SciPost Phys. Lect. Notes, 5 (2018). Code available from https://github.com/tenpy/tenpy.

[30] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar, Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits, Phys. Rev. Lett. **121**, 220502 (2018).

[31] N. Ezzell, B. Pokharel, L. Tewala, G. Quiroz, and D. A. Lidar, Dynamical decoupling for superconducting qubits: A performance survey, Phys. Rev. Appl. **20**, 064027 (2023).

[32] E. Van Den Berg, Z. K. Minev, and K. Temme, Model-free readout-error mitigation for quantum expectation values, Phys. Rev. A **105**, 032620 (2022).

[33] J. Tindall and M. Fishman, Gauging tensor networks with belief propagation, SciPost Phys. **15**, 222 (2023).

[34] H. N. Phien, I. P. McCulloch, and G. Vidal, Fast convergence of imaginary time evolution tensor network algorithms by recycling the environment, Phys. Rev. B **91**, 115137 (2015).

[35] M. P. Zaletel and F. Pollmann, Isometric tensor network states in two dimensions, Phys. Rev. Lett. **124**, 037201 (2020).

[36] G. Evenbly, Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops, Phys. Rev. B **98**, 085155 (2018).

[37] Y. Kim, A. Eddins, S. Anand, K. Xuan Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme *et al.*, Evidence for the utility of quantum computing before fault tolerance, Nature **618**, 500 (2023).

[38] P. D. Nation and M. Treinish, Suppressing quantum circuit errors due to system variability, PRX Quantum **4**, 010327 (2023).