

Efficient Tensor Network Simulation of IBM's Eagle Kicked Ising Experiment

Joseph Tindall^{1,*}, Matthew Fishman,¹ E. Miles Stoudenmire¹, and Dries Sels^{1,2}

¹Center for Computational Quantum Physics, Flatiron Institute, New York, New York 10010, USA

²Center for Quantum Phenomena, Department of Physics, New York University, 726 Broadway, New York, New York 10003, USA

 (Received 7 September 2023; accepted 25 November 2023; published 23 January 2024)

We report an accurate and efficient classical simulation of a kicked Ising quantum system on the heavy hexagon lattice. A simulation of this system was recently performed on a 127-qubit quantum processor using noise-mitigation techniques to enhance accuracy [Y. Kim *et al.*, *Nature*, 618, 500–5 (2023)]. Here we show that, by adopting a tensor network approach that reflects the geometry of the lattice and is approximately contracted using belief propagation, we can perform a classical simulation that is significantly more accurate and precise than the results obtained from the quantum processor and many other classical methods. We quantify the treelike correlations of the wave function in order to explain the accuracy of our belief propagation-based approach. We also show how our method allows us to perform simulations of the system to long times in the thermodynamic limit, corresponding to a quantum computer with an infinite number of qubits. Our tensor network approach has broader applications for simulating the dynamics of quantum systems with treelike correlations.

DOI: [10.1103/PRXQuantum.5.010308](https://doi.org/10.1103/PRXQuantum.5.010308)

I. MAIN

A. Introduction

Quantum computers are fundamentally noisy in nature and incur errors with each operation. To remedy noise and make the computer run as desired, one option could be quantum error-correcting codes. In spite of significant advances in quantum technologies over the last decade, this type of error correction is currently out of reach. As such, huge efforts have been devoted to finding out whether current noisy quantum technologies could provide a practical advantage over classical computers without error correction. The emphasis here is on practicality, since expectation values of local observables in large classes of shallow quantum circuits can formally be computed in polynomial time on a classical computer [1,2] and so can the output distribution of a noisy random quantum circuit [3]. The latter refutes the claim of violation of the extended Church-Turing thesis in Ref. [4], and makes it exceedingly unlikely any such violation can be achieved through noisy quantum computation without error correction. Noise can also have a similar effect as limiting entanglement,

allowing noisy quantum computers to be simulated by tensor network methods [5–7]. While such works put constraints on the power of noisy quantum devices, they also show that it may be difficult to classically simulate them in practice when the noise is sufficiently small. At the same time, error-mitigation techniques have been proposed to further improve the accuracy of noisy quantum devices by classical postprocessing [8,9]. With these techniques one can construct a better classical (biased) estimator for the desired noiseless result by paying the overhead of having to run several noisy quantum circuits.

The power of noise-mitigation techniques was recently demonstrated in quantum simulations of the dynamics of a two-dimensional (2D) transverse-field Ising model on a 127-qubit heavy hexagon lattice [10] (see Fig. 1). Expectation values of a number of observables were extracted after a few Floquet cycles using zero-noise extrapolation techniques [8]. The experimental results were found to be much more accurate than several classical tensor network approaches applied to the same problem, even when the tensor network methods utilized significant computational resources.

In this work we adopt a tensor network approach that respects the qubit connectivity of the heavy hex lattice to simulate the dynamics of this kicked Ising model on systems of various sizes. Our tensor network ansatz is approximately contracted with belief propagation, which works best when correlations in the system remain “treelike.” By quantifying these correlations we

*jtindall@flatironinstitute.org

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

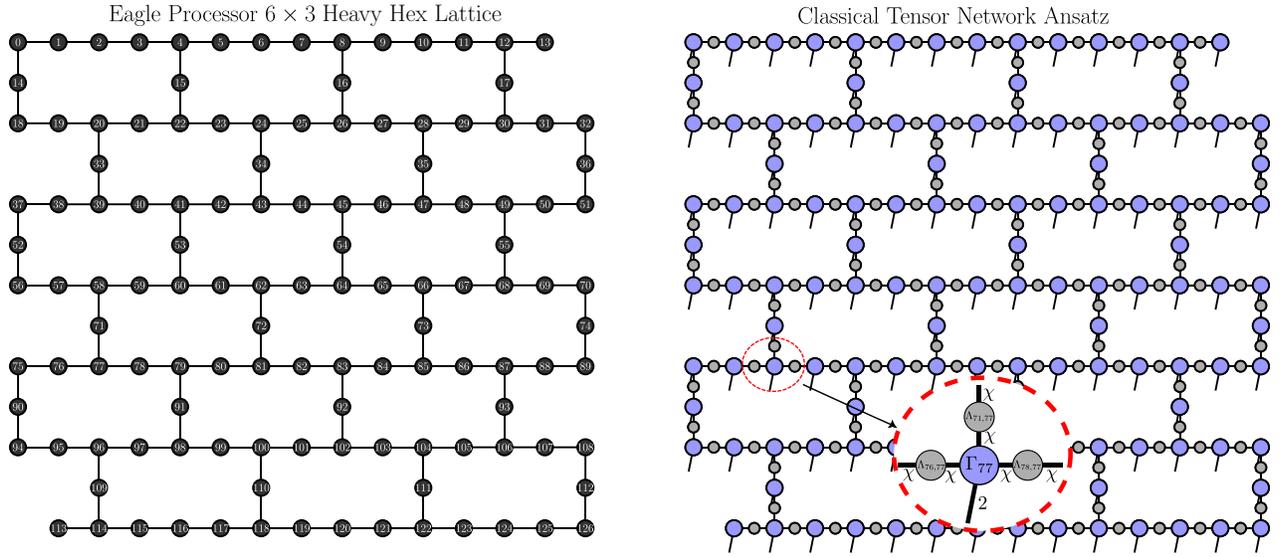


FIG. 1. Left: structure of the Eagle quantum processor, which consists of a 6×3 heavy hexagon lattice with two additional qubits (113 and 13) added to the bottom left and right corners of the lattice. Right: tensor network structure used for our simulations of heavy hex lattices, with the network structure directly reflecting the lattice. On-site tensors Γ_v are colored in blue and possess physical, uncontracted, indices of dimension 2 (represented by their dangling legs) and virtual indices of dimension χ (represented by the edges of the network), which are shared with neighboring tensors. Positive, diagonal bond tensors Λ_e live on the edges e between the site tensors and are colored in gray.

explicitly show how this assumption becomes increasingly valid as the lattice size increases. This allows our method to achieve results, for the 127-qubit problem, to a much higher degree of accuracy than previously reported classical results or the error-mitigated quantum computer. Even at large circuit depths where there are no exact results to benchmark against, we utilize extensive MPS calculations and boundary MPS approaches to provide substantial evidence that our results are still highly accurate and the correlations are treelike. Finally, we conclude by showing how our method can be used to accurately simulate the thermodynamic limit of the problem to long times, and therefore can simulate a high-depth quantum circuit involving an infinite number of qubits. Our work demonstrates the effectiveness of a belief propagation tensor network approach for solving many-body dynamics problems. We anticipate our chosen methodology will find success and serve as a benchmark when applied to problems with locally treelike correlations and limited entanglement.

B. Model and ansatz

Our focus here is on the dynamics of the Trotterized kicked transverse-field Ising model given by the unitary

$$U(\theta_h) = \left(\prod_{\langle v, v' \rangle} \exp\left(i\frac{\pi}{4} Z_v Z_{v'}\right) \right) \left(\prod_v \exp\left(-i\frac{\theta_h}{2} X_v\right) \right), \quad (1)$$

where Z and X denote Pauli operators and $\langle v, v' \rangle$ indicates that v and v' are neighbors on the corresponding lattice. The lattice we are concerned with is that of the “heavy hex” lattice, which corresponds to a hexagonal lattice decorated with additional qubits along the edges (see Fig. 1). The dynamics of this model was recently simulated on the IBM Eagle quantum processor [10], which corresponds to a lattice of 6×3 heavy hexagons plus two additional qubits.

In order to simulate this system on a classical computer, we adopt a tensor network approach that respects the qubit connectivity of the heavy hex lattice (see Fig. 1). We fix a maximum amount of entanglement in the system by limiting the bond dimension χ of the network: each application of $U(\theta_h)$ will double the bond dimension of the network and thus it is necessary to limit the bond dimension and perform truncations. We then evolve our tensor network state (TNS) by application of the gates in $U(\theta_h)$ under the belief propagation (BP) approximation [11–13]; referring to the resulting TNS as a BP-approximated TNS. Unless otherwise stated, we also extract expectation values from the TNS using belief propagation. Explicit details of our BP-based method are provided in Sec. II and in Ref. [14]. The BP method is fully controlled on trees but incurs a potentially small but uncontrolled approximation when there are loops in the network; the error from this approximation is, in general, smaller for larger loops (see Sec. II for further discussion on this). Here, we demonstrate that for a sufficiently large lattice, even at significant circuit depths, the correlations in this model remain

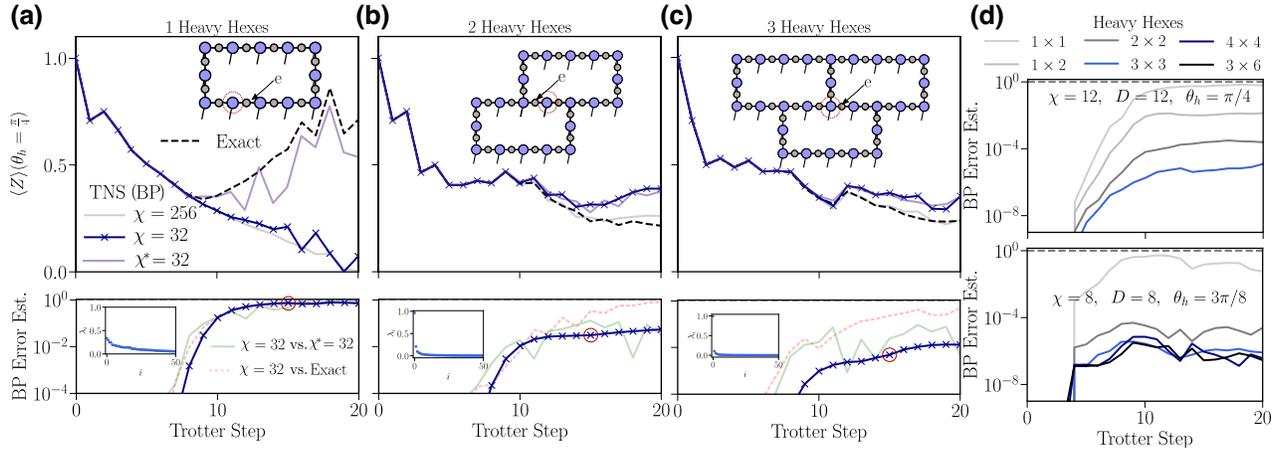


FIG. 2. Dynamics of the kicked Ising model on small heavy hex lattices of varying size. (a)–(c) Results from the BP-evolved TNS for several bond dimensions and systems of one to three heavy hexes are compared to exact state vector solutions at $\theta_h = \pi/4$. The results labeled with χ^* are obtained by computing expectation values of the BP-evolved TNS using exact contraction, while the results labeled with χ are obtained by computing expectation values of the BP-evolved TNS using BP contraction. In both cases, the states are evolved by applying gates using the BP approximation. Top plots show dynamics of the magnetization on the indicated (red ring) site, bottom plots show the BP error estimate [based on the spectrum of the edge environment—see Fig. 7 and Eq. (12)] for the $\chi = 32$ TNS along the indicated edge e versus the Trotter step. The dotted faded red line shows the relative error between the $\chi = 32$ TNS magnetization approximated by BP and the exact magnetization while the faded green line shows the relative error between the $\chi = 32$ TNS magnetization approximated by BP and the magnetization obtained by contracting the same TNS exactly. Insets in the bottom plots show the first 50 singular values of the edge environment after 15 Trotter steps. (d) BP error estimate approximated using a boundary MPS contraction scheme (see Appendix C for details) of the TNS for $n \times m$ lattices of n rows and m columns of heavy hexes. (Top) TNS with $\chi = 12$ and a boundary MPS contraction scheme with maximum MPS bond dimension $D = 12$ at $\theta_h = \pi/4$. (Bottom) TNS with $\chi = 8$ and a boundary MPS contraction scheme with maximum MPS bond dimension $D = 8$ at $\theta_h = 3\pi/8$.

“treelike” in the sense of the BP approximation giving very accurate results. Let us present these results.

C. Results

We start by considering lattices with a small number of heavy hexagons, where an exact state vector simulation is possible and our method can be directly benchmarked (see Fig. 2). Specifically, we compute the dynamics of the on-site magnetization for $\theta_h = \pi/2$ and lattices consisting of 1, 2, and 3 heavy hexagons, respectively. We also compute the separability of the “edge environment” of the BP-approximated TNS along a chosen edge (see Sec. II E for a definition). This environment corresponds to the contraction of the TNS down to the given edge and when it is completely separable the belief propagation assumption is exact. The separability thus gives us an estimate of the error stemming from the BP approximation.

For short-circuit depths $n < 6$ our method gives perfect agreement with the exact simulation and the edge environments are all completely separable because the light cone of the circuit does not reach around the loops. For larger circuit depths there is deviation between the exact dynamics and the BP-approximated TNS dynamics. This can be explicitly characterized by a decrease in the separability of the edge environments, since the BP method

approximates the environments used to perform the gate evolution as outer products of environments coming from incoming edges of a region of the system. Our small lattice simulations demonstrate something quite remarkable: as the number of heavy hexagons increases, the BP approximation at fixed χ improves significantly and the edge environment becomes highly separable even out to 20 Trotter steps. Moreover, in Fig. 2(d) we use boundary MPS (see Appendix C for a description of the method) to show that this increased separability persists with increasing system sizes.

The accuracy of belief propagation does tend to improve with increasing system size (for instance, belief propagation on a finite ring directly gives the exact results of the thermodynamic limit, independent of system size) and may go some way towards explaining the increased accuracy of the BP simulation as we increase the number of heavy hexes. Nonetheless, the heavy hex lattice still has finite loops in the thermodynamic limit and thus our results are indicative of something further: a level of interference between the loops in the lattice as we increase the system size. We note that Ising models have been observed to display some slow thermalization and confinement properties under a quantum quench [15,16] and we believe a similar effect is occurring here.

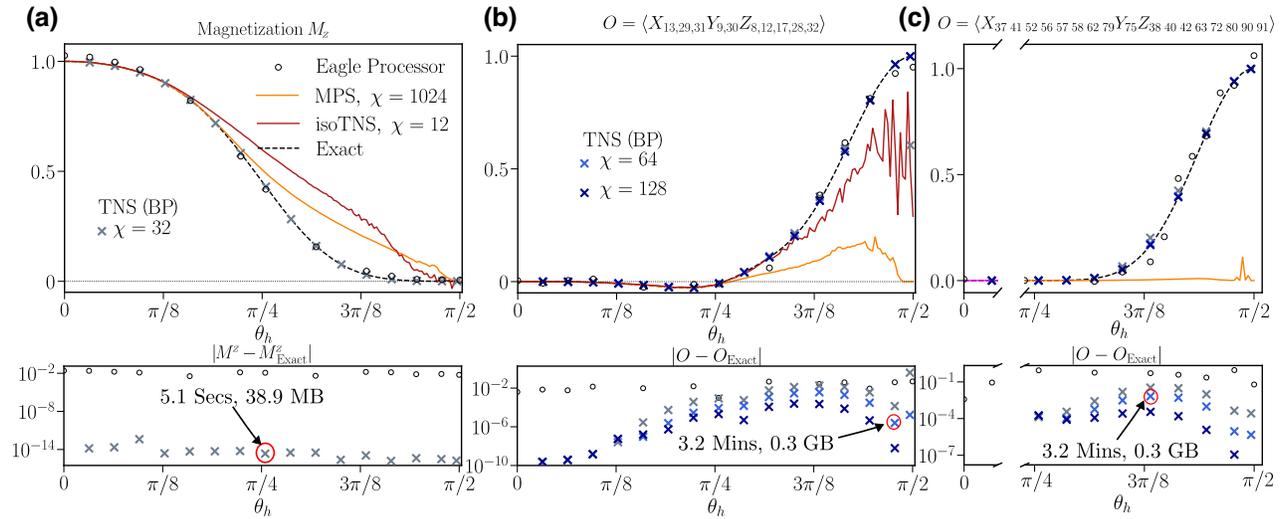


FIG. 3. Comparison for classically verifiable systems of our BP-approximated tensor network state approach to simulating the dynamics of the kicked transverse field Ising model on a heavy hex lattice versus the Eagle quantum processor and alternative tensor network methods. Expectation values with respect to the state $|\psi(\theta_h, 5)\rangle$ [i.e. following five Trotter steps of the dynamics of the model—see Eqs. (1) and (3)] are plotted, alongside exact results determined from light-cone simulations. (a) Average magnetization. (b) Weight-10 observable. (c) Weight-17 observable. The bottom plots show errors defined as the absolute difference between the simulation result and the exact result. For some data points the error from our TNS simulation is too small to fit on the scale of the plot and so these points are not marked. Circled, annotated points denote, for a given θ_h , the memory required to store the state of the system at the given bond dimension χ and the walltime associated with performing the simulation and calculating the relevant observable on a Macbook M1 Pro.

We now consider the 127-qubit heavy hexagon lattice, which corresponds to that of the IBM Eagle processor and we compare our method to the experimental quantum simulation results first from Fig. 3 of Ref. [10]. Our results on smaller system sizes will be useful in explaining the accuracy of our results here. In Fig. 3 we overlay the quantum simulation results with that of our BP-approximated TNS dynamics shown as cross symbols. Here expectation values are measured after five Trotter steps and exact results, based on brute-force light-cone simulation techniques, are available to allow us to directly assess the errors. The tensor network state (TNS) and gate-evolution methods we use simulate the full 127-qubit system and result in highly accurate expectation values. In Fig. 3(a) we compute the expected value of the average single-site magnetization and show that we can obtain an accuracy [17] of approximately 10^{-14} with a simulation that runs in less than 10 s on a laptop computer. Importantly, even for Figs. 3(b) and 3(c), where we consider higher weight observables, which could be more strongly affected by loop correlations, we are still able to calculate these observables to a remarkable accuracy using our BP-approximated TNS. Specifically, we obtain values of these higher weight observables to orders of magnitude better accuracy than the quantum processor with a simulation that takes less than 4 min (for these observables) to run on a laptop and a state that takes up, at most, 0.3 GB of memory. The remarkable accuracy we are able to achieve corroborates with our

earlier analysis of the error of the BP approximation as a function of system size (see Fig. 2), where we found that the BP error decreases as we increase the system size for this model and lattice.

Turning next to larger numbers of Trotter evolution steps we show results in Fig. 4 for properties also computed by the quantum processor. For the $n = 6$ Trotter-step simulation [Fig. 4(a)], where a weight-17 observable is measured, exact results are now available [18] and our BP-approximated TNS results at $\chi = 500$ are within 10^{-4} of these results for all values of θ_h we plot. For $n = 20$ [Fig. 4(b)] exact data is currently unavailable and we push to sufficiently large bond dimensions to perform an extrapolation of the results from BP-approximated TNS to infinite bond dimension; demonstrating the reliability of a linear extrapolation in $1/\chi$ for select θ_h in Fig. 4(c). We capture the known exact values at the Clifford points $\theta_h = 0$ and $\theta_h = \pi/2$. Notably, for this circuit in the region $\pi/8 \leq \theta_h \leq 3\pi/8$ there have been discrepancies between various classical methods and the quantum processor. In Appendix A, we discuss the different classical simulation methods further and show a comparison in Fig. 8 of the various results [10, 18–22].

Beyond the results we presented above giving evidence for the separable nature of the edge environments in this system, we present further evidence that the BP-approximated TNS method is highly accurate throughout the whole phase diagram in Figs. 4(d)–4(f). Specifically,

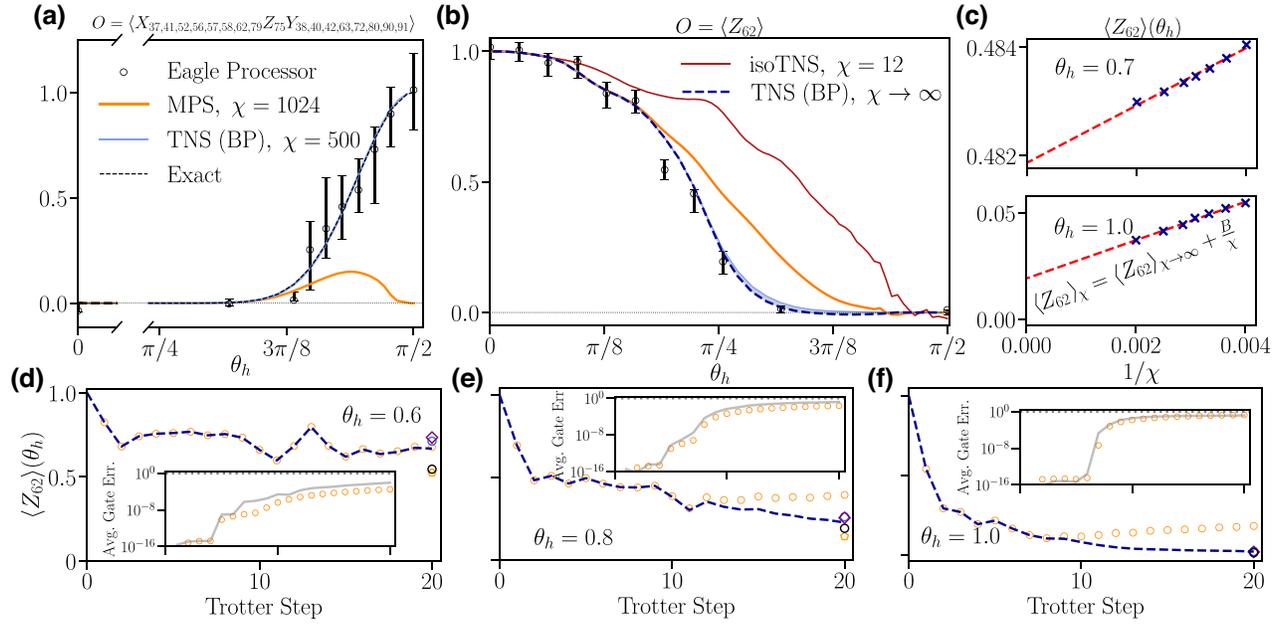


FIG. 4. Comparison for deeper circuits of our BP-approximated tensor network state approach to simulating the dynamics of the kicked transverse-field Ising model on the heavy hex lattice versus the Eagle quantum processor and alternative tensor network methods. Expectation values calculated following a number of Trotter steps of the dynamics of the model—see Eq. (1)—are plotted. (a) Weight-17 stabilizer after six steps of evolution. Here exact data is now available [18] and our BP data for $\chi = 500$ is within 10^{-4} of the exact result for all θ_h . (b) Weight-1 observable after 20 steps. The shaded region shows the difference between our finite $\chi = 500$ bond-dimension data and the data extrapolated to infinite bond dimension, where we believe the true answer lies. (c) Top and bottom plots show observables in (b) at $\theta_h = 0.7$ and $\theta_h = 1.0$, respectively, as a function of inverse bond dimension of the TNS. Red dashed lines represent a least-squares fit of the form $A + B/\chi$ taken on the data, and we take A to be the predicted value of the observable in the limit $\chi \rightarrow \infty$. Even in the limit $\chi \rightarrow \infty$ there will generally be some deviation from the exact result due to the BP approximation that we use for evolving the state and computing expectation values (see Sec. II). Our analysis of the errors due to BP for this system, however, suggest this deviation is likely to be very small. (d)–(f) Dynamics of $\langle Z_{62} \rangle$ using the BP-approximated TNS approach versus a MPS approach (with bond dimension 2500) with light-cone depth reduction (orange) for $\theta_h = 0.6, 0.8$, and 1.0 , respectively. Results from other methods at depth 20 are shown as black circles (Eagle processor [10]), blue circles (truncated Pauli strings [20,21]), purple diamonds (TNO [18]) and orange pentagons (MPO [19]). Inset shows average gate error from the MPS approach (pink circles) and absolute difference between the BP-approximated TNS and the MPS result (solid gray line).

we compute the dynamics at every Trotter step of $\langle Z_{62} \rangle$ for several θ_h and compare the BP-approximated TNS to our own MPS calculations as an independent check. Our MPS approach combines multiple nontrivial techniques: (i) utilizing light-cone depth reduction to calculate $\langle Z_{62} \rangle$ at every Trotter step, (ii) using a higher bond dimension than that in Ref. [10], and (iii) implementing an improved site ordering to lower the entanglement and gate error. We find that the difference between the BP-approximated TNS and MPS is directly correlated with the error from the MPS method and that both methods agree closely when the error in the MPS method is itself small. When the MPS error is small, one can consider MPS to be exact as it makes no uncontrolled approximations. The fact the BP-based method agrees with the MPS method when MPS exhibits very small errors suggest the BP error is also minimal. This is clearest for $\theta_h = 0.6$ in Fig. 4(d) where we are able to push our MPS simulations to a bond dimension large enough for the average gate error to stay below 10^{-4} . At all times for $\theta_h = 0.6$, there is clear agreement between BP-approximated TNS

and MPS, yet disagreement at depth 20 versus the other methods shown [18–22]. This agreement is only possible if the state possesses treelike correlations and thus reinforces our earlier results on the general accuracy of the BP approximation for the dynamics of this system on this lattice. For larger θ_h , in particular the $\theta_h = 1.0$ results shown in Fig. 4(f), the BP results agree with the new MPS results until about step 10 where the MPS error starts to grow. We can be confident that the discrepancy there is due to the MPS method, not BP, because MPS is a controlled method that self-reports a significant error for larger steps and because the BP agrees with the result of approximately 0 at step 20, which is predicted by a range of classical methods [18–22] due to the system being ergodic in this regime.

D. Dynamics of the infinite heavy hex lattice

One of the powerful features of tensor network methods is their ability to simulate *infinite* lattice models

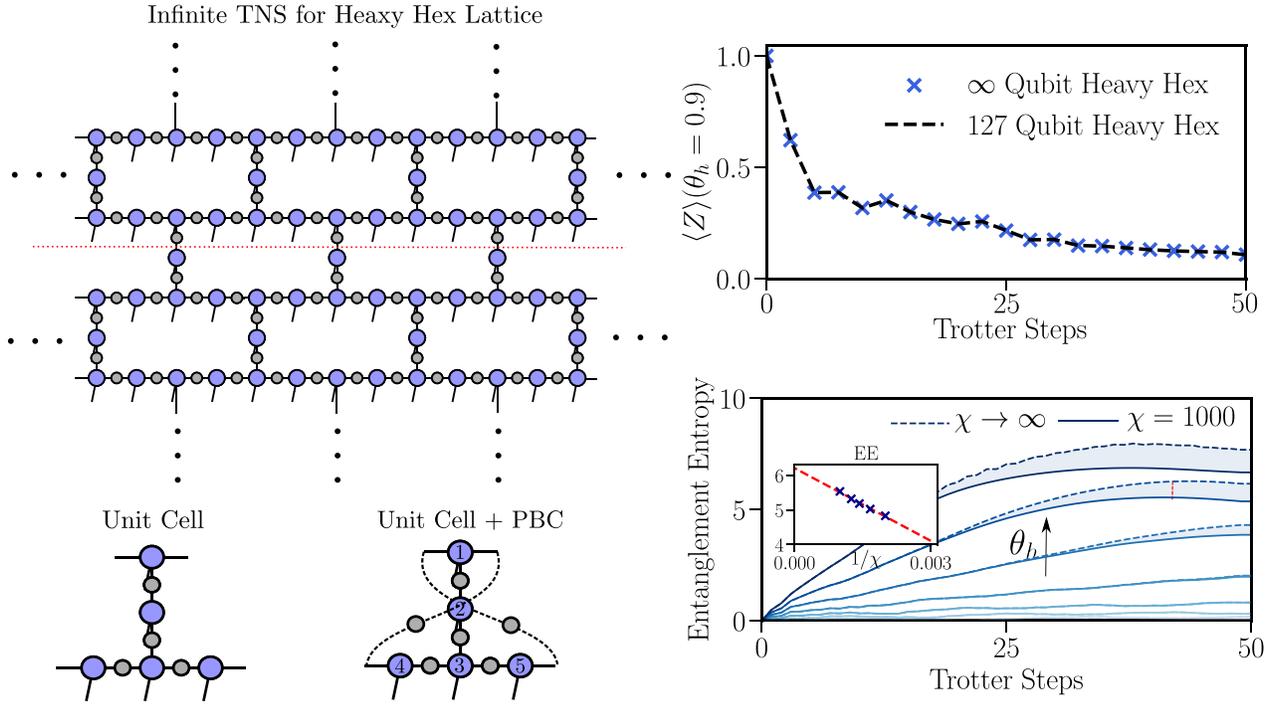


FIG. 5. Tensor network state for the infinite heavy hex lattice. The unit cell is a five-site tensor network. By adding in appropriate periodic boundary conditions and simulating the kicked Ising model with the BP-approximated TNS method we recover results for simulation of the infinite lattice with the BP-approximated TNS method. (Top right) Dynamics of $\langle Z_{62} \rangle$ for $\theta_h = 0.9$ simulated using the BP-approximated TNS method at bond dimension $\chi = 400$. Crosses correspond to the dynamics of $\langle Z_3 \rangle$ on the periodic boundary condition (PBC) unit cell (where site 3 is marked) with $\chi = 400$ corresponding to an infinite heavy hex lattice while the dashed line corresponds to the dynamics of $\langle Z_{62} \rangle$ on the 127 qubit heavy hex lattice with $\chi = 400$. (Bottom right) Dynamics of the bipartite entanglement entropy density s [see Eq. (2) for the definition], calculated using the infinite BP-approximated TNS method and extrapolated to infinite bond dimension. Different curves correspond to different values of θ_h , ranging in steps of 0.1 from 0.1 to 0.9. The partition is shown by the red dotted line on the infinite lattice. The shaded area shows the difference between the extrapolated result (dotted line) and the finite $\chi = 800$ result. The inset shows an example extrapolation in $1/\chi$ for $\theta_h = 0.8$ after 40 Trotter steps.

as long as they possess some form of translational invariance [23–25]. Here we present results on the dynamics of the kicked transverse-field Ising model on an infinite heavy hexagon lattice, corresponding to a quantum computer with an infinite number of qubits. Again, we approximate the dynamics and take expectation values using the BP approximation. Given the evidence that we have presented on the accuracy of the BP approximation for the finite case of this system, especially for larger system sizes, we expect that these results are highly accurate.

For the infinite heavy hex lattice there is a five-site unit cell, which can be tiled to produce the infinite lattice (see Fig. 5). It can be shown that simulating a periodic system on the unit cell using belief propagation corresponds to simulating the infinite lattice under the belief-propagation approximation, where the sites of the periodic system constitute a unit cell of the infinite system [14]. This is analogous to the standard approach to simulating infinite systems with the simple update tensor network method [26]. Therefore, we take the single unit cell, impose periodic boundary conditions, and present results for the belief-propagation approximated dynamics. Figure 5

illustrates this idea, showing the dynamics of the magnetization of the infinite system compared to the expected magnetization of a representative site near the center of the finite system. The extremely close agreement between the magnetization of the infinite and finite heavy hex lattices provides strong evidence that boundary effects are minimal in the 127-qubit model and the results are already very close to those of the thermodynamic limit.

In Fig. 5 we also show the time-dependence of the bipartite entanglement entropy per edge s across a bipartition of the infinite lattice. Our infinite BP-approximated TNS method gives us an estimate of this quantity via the spectrum $\{\lambda_i\}$ of the bond tensors along the edges separating the partitions:

$$s = \sum_{i=1}^{\chi^2} \lambda_i^2 \log_2(\lambda_i^2), \quad (2)$$

which requires summing over the spectrum of only one of the bond tensors due to the fact that they are all identical along the cut being made. We observe that the

entanglement in the system shows a sharp linear growth at short times before slowing down significantly and potentially saturating over a large time scale. This long-lived plateau is consistent with studies of the decay of the magnetization in a smaller system [22], which suggests that the time-to-decay scales exponentially with the inverse of θ_h . This long-time slow growth means that we can accurately simulate the infinite quantum processor up to large circuit depths for smaller values of $\theta_h \lesssim \pi/4$. For larger values of θ_h , the entanglement grows sufficiently quickly that, with our current resources, we are unable to accurately determine the entanglement entropy in the system beyond approximately 25 Trotter steps.

E. State ansatz

Our ansatz for the wave function of the system is a tensor network, which directly reflects the “heavy hex” qubit connectivity of the processor (see Fig. 1). The physical properties of a tensor network are invariant under a gauge symmetry corresponding to the insertion of any invertible matrix G_e and its inverse G_e^{-1} on any contracted, internal bond (or edge) e of the network. We take advantage of this symmetry to keep the tensor network in the *Vidal gauge* [26,27]. This gauge corresponds to the choice of positive, diagonal bond tensors Λ_e residing on the edges of the network and the on-site tensors Γ_v of the network obeying certain isometric properties [see Eqs. (5) and (6)]. These isometric properties are important for maintaining accuracy during the evolution of the network. We use $|\psi(\theta_h, n)\rangle$ to denote the TNS of the system after $n \geq 1$ applications of $U(\theta_h)$, i.e.,

$$|\psi(\theta_h, n)\rangle = \left(\prod_{i=1}^n U(\theta_h) \right) |\psi(0)\rangle = U^n(\theta_h) |\psi(0)\rangle, \quad (3)$$

where $|\psi(0)\rangle$ is the initial state of the system. We use the same initial state as in Ref. [10]: $|\psi(0)\rangle = |\uparrow\uparrow\dots\uparrow\rangle$. The single-site X rotations in $U(\theta_h)$ can be applied to $|\psi(\theta_h, n)\rangle$ exactly and the two-site gates are applied approximately using the simple update [26] procedure (see Sec. II), which involves truncating the internal indices of the TNS to keep them less than or equal to a prescribed maximum bond dimension χ . Following a single Trotter step, the tensor network is regauged using belief propagation, a well-established statistical inference algorithm, which can be formulated for tensor networks [12], which we find improves the accuracy of the simple update procedure [14,28]. Regauging before applying each gate with a simple update would be most accurate and would be equivalent to performing each gate application with environments computed from the BP fixed point, however in practice we find that is too computationally expensive. Regauging after each Trotter step provides a good balance between accuracy and speed. Belief propagation allows us

to rapidly determine the necessary transformation matrices by performing *message passing* over the network representing $\langle \psi(\theta_h, n) | \psi(\theta_h, n) \rangle$. The use of belief propagation as a method to efficiently “regauge” a tensor network was recently formulated by some of the current authors in Ref. [14] and is closely related to other known gauging methods [28–31]. We emphasize that the results presented here could have been achieved with those known gauging methods.

F. Measuring expectation values

We measure single-site expectation values of the gauged state $|\psi(\theta_h, n)\rangle$ using a rank-one approximation for the environments of local regions of the network (see Sec. II for more details). Such a method is only guaranteed to be exact in the limit of a tree network. Nonetheless, as demonstrated by our results here, the large loop structure of the heavy hexagon lattice and dynamics of the model means that the network is locally treelike and therefore amenable to such an approximation. In Ref. [10], specific higher-weight observables were also measured to highlight that nontrivial results can be obtained in the “regime of strong entanglement”—such observables can be difficult to accurately measure for loopy tensor networks. We can, however, exploit the Clifford properties of the circuit at $\theta_h = \pi/2$ to transform the problem of measuring higher-weight observables after n Trotter steps into one of measuring a single-site observable after evolution by n Trotter steps with the propagator $U(\theta_h)$ and n Trotter steps with the propagator $(U(\pi/2))^\dagger$. Specifically, the operator $U^n(\pi/2) Z_i (U^n(\pi/2))^\dagger$ is always a single Pauli string and its expectation with respect to $|\psi(\theta_h, n)\rangle$ can be obtained by further evolving $|\psi(\theta_h, n)\rangle$ with $(U(\pi/2))^\dagger$ and then measuring Z_i . For instance,

$$X_{13,29,31} Y_{9,30} Z_{8,12,17,28,32} = U^\delta \left(\frac{\pi}{2} \right) Z_{13} \left(U^\delta \left(\frac{\pi}{2} \right) \right)^\dagger, \quad (4)$$

meaning this observable can be calculated, with respect to $|\psi(\theta_h, 5)\rangle$, by measuring $\langle Z_{13} \rangle$ with respect to the state $(U^\delta(\pi/2))^\dagger |\psi(\theta_h, 5)\rangle$. We can reach this state straightforwardly by performing a 10 (5 + 5) Trotter step tensor network simulation. We use this “extended time evolution” method to measure all higher-weight observables presented in Ref. [10] and note that this procedure is entirely generic: any long Pauli string can be generated out of a single Pauli by application of a Clifford circuit. Naturally this extended time evolution further increases the entanglement we must deal with as each application of $U(\pi/2)$ will double the bond dimension if no truncation is performed. Thus an upper bound on the bond dimension χ_{\max} needed to evolve the system by n Trotter steps and the measure a string, which can be generated by n' applications of $U(\pi/2)$ to a single-site Pauli operator is $2^{n+n'}$. Generically, longer strings will require larger values of n' .

they form an isometric tensor obeying

$$\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} W_{v,v_i}^* \\ \vdots \\ W_{v,v_i} \end{array} \text{---} = \text{---} \quad , \quad (6)$$

where the right-hand side denotes the identity matrix and we have used the set $\{v_1, v_2, \dots, v_d\}$ to denote the d neighbors of vertex v . Having the TNS obey this property is important for maintaining accuracy when applying two-site gates and when taking expectation values. Below we detail the methods we use for performing these operations and then describe how we use belief propagation to maintain the Vidal gauge during our simulations. We emphasize that working in the Vidal gauge is not strictly necessary, and one can perform all of the same operations (gate application, computing expectation values, etc.) and get the same results without transforming to the Vidal gauge by just using message tensors found from performing belief propagation on the TNS in an *arbitrary gauge* [14]. This is why we opt to name our tensor network state as a ‘‘BP-approximated TNS’’ and will discuss further in the sections below.

B. The simple update procedure

To apply two-site gates to the tensor network state we adopt the simple update procedure [26]. The procedure is depicted diagrammatically in Fig. 6. For better efficiency, we use the ‘‘reduced tensor’’ variant of simple update [36,37] (not shown in Fig. 6). The simple update procedure can be performed on a TNS in an arbitrary gauge by working with the fixed point message tensors found from belief propagation on the TNS and updating the message tensor on the edge where the gate is applied with the bond matrix returned from the SVD procedure [14]. This is similar to previous work in Ref. [12] where message tensors were used for energy optimization.

C. Measuring expectation values within the Vidal gauge

In order to measure a single-site observable $\langle O_v \rangle$ on site v of our TNS we absorb the neighboring bond tensors onto the on-site tensor Γ_v and contract the result with its conjugate, inserting the single-site observable O_v along the physical index, which is being contracted over. For the example of a site with three neighbors this can be visualized as

$$\langle O_v \rangle \approx \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} \Gamma_v \\ | \\ O_v \\ | \\ \Gamma_v^* \end{array} \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \quad , \quad (7)$$

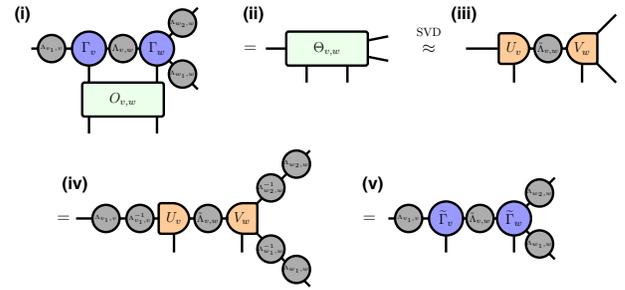


FIG. 6. Steps to perform a ‘‘simple update’’ on the edge of a TNS in the Vidal gauge. The example pictured is two neighboring sites of degree 2 (site tensor Γ_v) and 3 (site tensor Γ_w), respectively. (i)–(ii) The bond tensors, site tensors, and gate are combined into a single composite tensor $\Theta_{v,w}$. (ii)–(iii) A SVD is performed on $\Theta_{v,w}$ and singular values of the resulting bond tensor $\tilde{\Lambda}_{v,w}$ can be discarded in order to truncate the bond. (iii)–(iv) Resolutions of identity, using the original bond tensors, are inserted on the exposed edges. (iv)–(v) The inverse bond tensors are absorbed, resulting in the updated local tensors $\tilde{\Gamma}_v$ and $\tilde{\Gamma}_w$.

where we have defined the square of the bond tensors

$$\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \Lambda_{v,v_i}^2 \text{---} = \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \Lambda_{v,v_i} \text{---} \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \Lambda_{v,v_i} \text{---} \quad . \quad (8)$$

Equation (7) approximates $\langle O_v \rangle$ by treating the environment as a tensor product of environments coming from each of the neighbors of v . If the tensors of the network obey the Vidal gauge conditions [Eqs. (5) and (6)], this approximation is equivalent to computing the expectation value in an arbitrary gauge by using the fixed-point message tensors of belief propagation as approximations of the environments [14,38]. For networks that are locally treelike, this can provide very good approximations for local observables. We find evidence that this approximation holds very well for the model and lattice studied in this work. In the section below we detail how we use belief propagation to maintain the Vidal gauge during our simulations. This helps maintain accuracy in the simple update procedure and when taking expectation values.

D. Belief propagation on a tensor network state

At the heart of our method for maintaining the gauge properties of our tensor network state is belief propagation (BP). BP is a well-established technique for approximating the marginals of the probability distributions of graphical models [39] and has recently gained interest in the context of contracting tensor networks [11–13,31].

To perform belief propagation on a TNS $|\psi\rangle$ in the Vidal gauge we first absorb the square roots of the bond tensors onto the Γ_v tensors. For instance, for the example of a

tensor with three neighbors we define

$$\begin{array}{c} \Lambda_{v_1,v}^{1/2} \\ \Lambda_{v_2,v}^{1/2} \\ \Lambda_{v_3,v}^{1/2} \end{array} \Gamma_v = T_v \quad (9)$$

We then form the closed network, which represents the square norm $\langle \psi | \psi \rangle$ of $|\psi\rangle$. This “norm” network consists of the on-site tensors $\mathcal{T}_v = \sum_{s_v} (T_v^{s_v})(T_v^{s_v})^*$, where the summation is over the external indices s_v of the tensors T_v and $(T_v)^*$ (though in practice we keep the tensors separate for efficiency).

We next define a series of “message tensors” over the norm network, with M_{v,v_i} denoting the message tensor directed along the edge from \mathcal{T}_v to its neighbor \mathcal{T}_{v_i} . The indices of M_{v,v_i} match the indices shared by tensors \mathcal{T}_v and \mathcal{T}_{v_i} . Here, the direction of the edge is important and, generally, $M_{v,v_i} \neq M_{v_i,v}$. A set of self-consistent equations is defined for the message tensors:

$$M_{v,v_i} = \left(\prod_{j \in \{1,2,\dots,d\} \setminus \{i\}} M_{v_j,v} \right) \mathcal{T}_v, \quad (10)$$

where the product runs over all d neighbors of v excluding v_i and multiplication of two tensors implies a contraction over any common indices they share. This equation can be expressed diagrammatically as

$$\begin{array}{c} M_{v_1,v} \\ \dots \\ M_{v_{i-1},v} \\ \mathcal{T}_v \\ M_{v_{i+1},v} \\ \dots \\ M_{v_d,v} \end{array} = M_{v,v_i} \quad (11)$$

Initializing the message tensors, one can iterate these equations in an attempt to converge them. The converged messages then form a rank-one approximation of the exact environment for a given tensor \mathcal{T}_v , and can be used to approximate expectation values of the TNS. The procedure for taking an expectation value in Eq. (7) using a state in the Vidal gauge is equivalent to taking the same state in an arbitrary gauge and approximating the environment surrounding the site with the BP message tensors [14,38]. The more treelike the network, the better the approximation.

Importantly, the BP message tensors can also be used to directly define a gauge transformation, which, when applied to a TNS, brings it into the Vidal gauge and guarantees the satisfaction of Eqs. (5) and (6) along all edges of

the network. A detailed description of this method, including extensive benchmarking and discussion of its relation to other gauging methods, is given in Ref. [14].

In our simulations here we perform belief-propagation gauging after every Trotter step. This is to maintain accuracy in these procedures. We should note that for shorter depth circuits, such as those simulated in Fig. 3, gauging after every Trotter step does not have a significant affect on accuracy. The circuit here is not deep enough to significantly alter the isometric condition in Eq. (6) and affect the simple update procedure. For longer depth circuits, such as the one run in Fig. 4(b), gauging every Trotter step is important and not performing gauging during the simulation leads to a significant loss in accuracy.

We also always gauge the state before taking expectation values (which, as we have discussed, is equivalent to computing the expectation values with fixed point BP message tensors). Even for the shorter circuits in Fig. 3, not performing gauging prior to taking an expectation value can noticeably affect the accuracy of the result—unless the bond dimension used is large enough for the simulation to be exact and so the simple update procedure preserves the gauge properties of the TNS.

E. Estimating the error of BP for general tensor networks

In order to quantify the error that belief propagation makes when approximating contractions of the network we compute the $\chi^2 \times \chi^2$ “edge environment” associated with cutting the norm network along a given edge. The edge environment has previously been used in other contexts, such as improving the performance of periodic MPS methods [40] and fixing the gauge and performing truncations of general tensor networks [41]. Here we use it to define a measure of the error of belief propagation. Specifically, we choose an edge of the norm network of our tensor network state and split the corresponding index (which is formed from the product of the bra and ket indices of that edge)—contracting the network down to a single matrix. This process is pictured in Fig. 7 for an example network.

Belief propagation essentially corresponds to assuming this edge environment is rank-one along every edge of the network, i.e., that the spectrum of the positive diagonal matrix of singular values Λ is $\{\lambda_1, \lambda_2, \dots, \lambda_{\chi^2}\} = \{1, 0, \dots, 0\}$. We can thus estimate the error from belief propagation along an edge e as

$$1 - \sqrt{\frac{\lambda_1^2}{\sum_{i=1}^{\chi^2} \lambda_i^2}}, \quad (12)$$

where the quantity inside the square root is known as the “index of separability” and is an established measure of the separability of a matrix based on a singular value decomposition [42]. Although the singular values are dependent

on the gauge of the tensor network, we fix to the symmetric gauge [14] when calculating them for consistency. This is the quantity we compute in Fig. 2 for the pictured networks and given edges. We also include examples of spectra, which correspond to a variety of separabilities in the insets. The BP error measure defined in Eq. (12) is in the same spirit as the “cycle entropy” defined in Ref. [41]. In general for this model, we find the choice of edge is unimportant and does not qualitatively change the results we observe.

It is worth making a further comment on the BP error in Eq. (12). In general, computing it is very costly due to the need to contract the full norm network, even if it is contracted approximately. One can observe, however, that generically the error will decrease exponentially in the smallest loop size of the lattice. This is most straightforwardly seen by considering a norm network, which is a single translationally invariant periodic ring (loop) of size l . Assuming the spectrum of the on-site transfer matrix \mathcal{T} is gapped then the BP error of the ring will scale as $\mathcal{O}(\exp(-cl))$ for some constant c related to the correlation length. This scaling should be generic for a lattice with smallest loop size l and gapped loop correlations.

F. Computing resources and software packages

The code used to produce the numerical results in this paper was written using the **ITensorNetworks.jl** package [35]—a general purpose and publicly available Julia [43] package for manipulating (gauging, contracting, partitioning, evolving, etc.) tensor network states of arbitrary geometry. It is built on top of **ITensors.jl** [44], which provides the basic tensor operations. Code is available in the current version of **ITensorNetworks.jl** for performing belief propagation, gauging, and the simple update procedure on arbitrary tensor network states. An example script is also included for specifically simulating the model in this paper with our BP-approximated TNS approach. All data was produced using a single node of Flatiron Institute’s Rusty computing cluster. Timings and memory usage quoted for Fig. 3, however, are based on the same code being run on a Macbook M1 pro. The tensor network diagrams in this paper were produced using the publicly available package **GraphTikz.jl** [45], a general-purpose Julia package for visualizing graphs, including tensor networks. Data for the 127-qubit simulations is currently available at <https://github.com/JoeyT1994/BP-TNS-Data>.

ACKNOWLEDGMENTS

J.T., M.F., M.S., and D.S. are grateful for ongoing support through the Flatiron Institute, a division of the Simons Foundation. D.S. was supported by AFOSR: Grant No. FA9550-21-1-0236. We would like to acknowledge Michael Zalatel, Sajant Anand, Garnet Chan, Andy Millis, Antoine Georges, and Bo Xiao for insightful discussions

and helpful comments on the draft. We would also like to thank Michael Zalatel, Sajant Anand, Garnet Chan, Tomislav Begušić, and Haijun Liao for sharing their data with us.

APPENDIX A: COMPARISON OF METHODS FOR CALCULATING Z_{62}

Here we compare data for the expectation value of $\langle Z_{62} \rangle$ after 20 Trotter steps from a range of classical methods and the quantum processor. The results are shown in Fig. 8. We include results at $\chi = 200$ and $\chi = 500$ from our BP-evolved TNS approach, as well as results from an extrapolation in $1/\chi$ to $\chi \rightarrow \infty$. The shaded region shows the difference between our finite $\chi = 500$ bond-dimension data and the data extrapolated to infinite bond dimension, where we believe the true answer lies. For smaller bond dimensions the BP-evolved TNS method overshoots the true results in the region $\pi/4 \lesssim \theta_h \lesssim 7\pi/16$, an artifact which is resolved by extrapolating in the bond dimension. Our extrapolated BP-evolved TNS results are in close agreement with results obtained in Ref. [21], which uses a combination of a BP-evolved TNS and a BP-evolved tensor network operator (TNO) to perform a mixed Schrödinger and Heisenberg evolution. Moreover, a method based on truncated Pauli strings [20,21], which approximates the Heisenberg evolution of the system and relies on a different type of approximation than tensor network methods and BP, also shows close agreement with our BP-evolved TNS approach and the aforementioned BP-evolved mixed TNS-TNO approach, although it slightly overshoots the value of $\langle Z_{62} \rangle$ for $5\pi/32 \lesssim \theta_h \lesssim 7\pi/32$ —which can be observed in Fig. 4(d) where we have numerically accurate results from MPS calculations.

Results in Fig. 8 are also shown for a matrix product operator (MPO)-based method of simulating the Heisenberg evolution of the system [19]. This approach, however, undershoots the value in the region $4\pi/16 \lesssim \theta_h \lesssim 5\pi/16$ —mapping the system to a 1D ansatz suffers from the drawback of requiring much larger bond dimensions than the more general tensor network-based approaches. The limitations of methods based on 1D ansatzes for this problem, like MPS and MPO, is especially clear when considering the MPS data originally reported in Ref. [10], along with the MPS results reported in this work in Fig. 4. We also include data from calculations based on a tensor network operator (TNO) approach [18], extrapolated to $\chi \rightarrow \infty$, which is evolved using a simple update (SU)—which is closely related to BP [14,38]—and then is contracted exactly to compute expectation values [46]. This method overshoots the true value of $\langle Z_{62} \rangle$ in the region $\pi/8 \lesssim \theta_h \lesssim 2\pi/8$ —which can be directly observed in Fig. 4(d). Finally, we provide data from a smaller, 31-qubit system calculation done in Ref. [22].

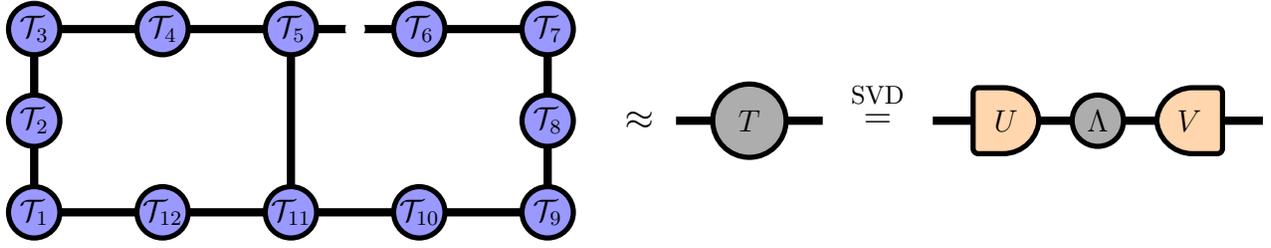


FIG. 7. Forming the edge environment from the norm network of a tensor network state. One of the edges e is split and all other indices of the network are contracted over, reducing the cut network to a single matrix where a singular value decomposition can be performed.

We also would like to point out work in Ref. [47], which used a dissipative mean-field theory to simulate the 127-qubit system out to short-circuit depths (approximately five Trotter steps) and obtained relatively accurate results—although with some noticeable deviations from the true result, which can be computed to an accuracy approximately 10^{-14} with our BP-evolved TNS method.

APPENDIX B: MPS CALCULATIONS

We performed MPS calculations for benchmarking in the verifiable regime in Figs. 4(d)–4(f). We implemented an ordering of the sites of the heavy hex lattice, which is distinct from that in Ref. [10]. Specifically, Fig. 9 shows the qubits of the Eagle processor numbered as 1 through 127, which then map directly to the sites 1 through 127 of the MPS we use. Two numberings are shown: our own

and the original ordering from Ref. [10]. We find that our own ordering results in slightly lower truncation errors when simulating the 127-qubit kicked Ising model at fixed MPS bond dimension χ . We break the two-site term in the propagator $U(\theta_h)$ down into a product of, at most, five commuting matrix product operators (MPOs) of bond dimension 2. The MPS is then evolved by successively applying these MPOs and after each application truncation is performed of the state down to a maximum bond dimension χ . The single-site gates are applied at the start of each Trotter step exactly. If implementing all two-site gates our decomposition of $U(\theta_h)$ involves 5 MPOs. In practice, however, we employ light-cone depth reduction (LCDR), which means that if there are n' Trotter steps left until we take a measurement on a specific site, we only include two-site gates in the MPOs, which are within the remaining light cone of the simulation. This means that for

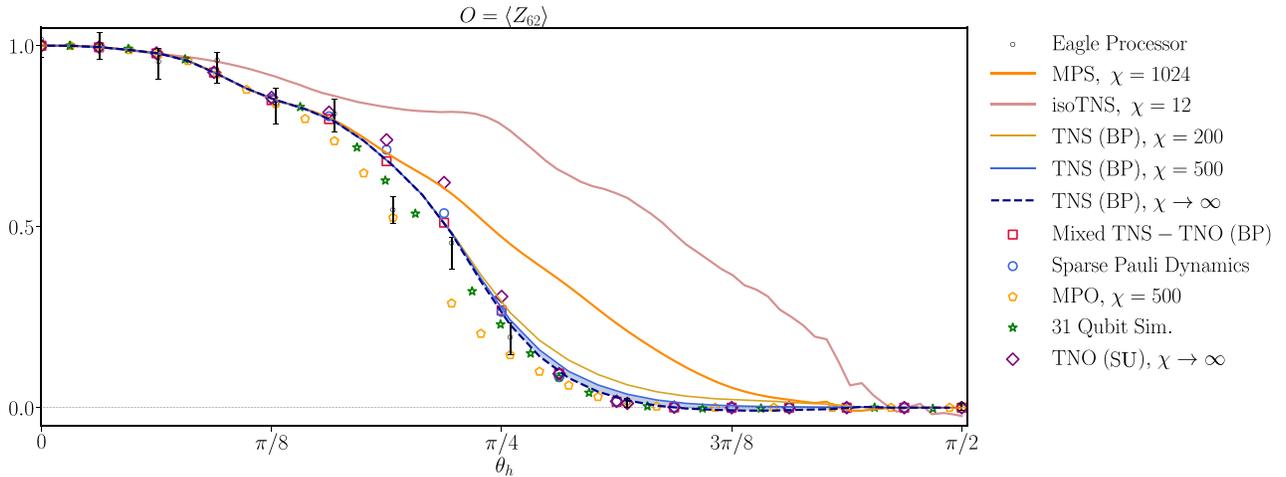


FIG. 8. Comparison of various approaches to simulating the kicked transverse-field Ising model on the 127-qubit Eagle processor geometry (Fig. 1). Approaches include a quantum processor (**Eagle Processor** [10]), **MPS** [10] and **isoTNS** [10] methods using Schrödinger evolution, a **TNS (BP)** approach using Schrödinger evolution and evolved with BP (this work), a **Mixed TNS-TNO (BP)** [21] approach using a combination of Schrödinger and Heisenberg evolution and evolved with BP, **Sparse Pauli Dynamics** [20,21], a **MPO** [19] approach using Heisenberg evolution, a 31-qubit full state simulation (**31 Qubit Sim.** [22]), and a **TNO (SU)** [18] approach using Heisenberg evolution and evolved with simple update, which is closely related to the BP approximation [14,38], and contracted exactly to compute expectation values.

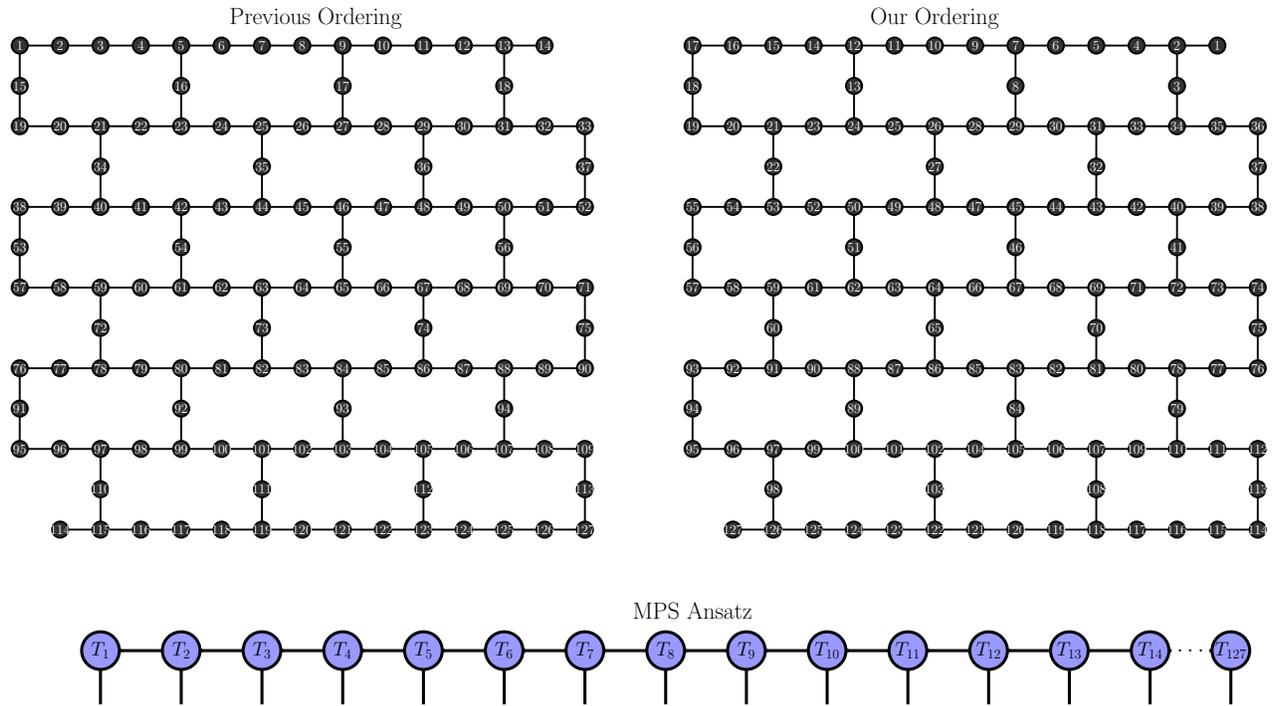


FIG. 9. (Top) Two possible mappings from the qubits of the Eagle processor to the sites of a MPS ansatz. The left diagram shows the ordering used in Ref. [10] while the right diagram shows the ordering we use to obtain the results in Figs. 4(d)–4(f), which we find leads to higher accuracy for a given MPS bond dimension. (Bottom) MPS ansatz with the site numbers corresponding to those in the above orderings.

a given θ_h in Fig. 4 we perform 20 separate simulations to get the most accurate MPS results for the value of $\langle Z_{62} \rangle$ after each number of steps n .

In order to approximate the error of this simulation we calculate the sum of the singular values discarded during the application of a single MPO, which we call ϵ_i , where i refers to the MPO being applied. The total approximate error during the n Trotter-step simulation [and plotted in

the insets of Figs. 4(d)–4(f)] is then [5]

$$E_n = 1 - \sum_{i=1}^N (1 - \epsilon_i)^{\frac{1}{N}}, \quad (\text{B1})$$

where the sum runs over all N MPO MPS applications (which may be less than $5n$ due to LCDR) during the simulation up to n Trotter steps. The error per gate that we

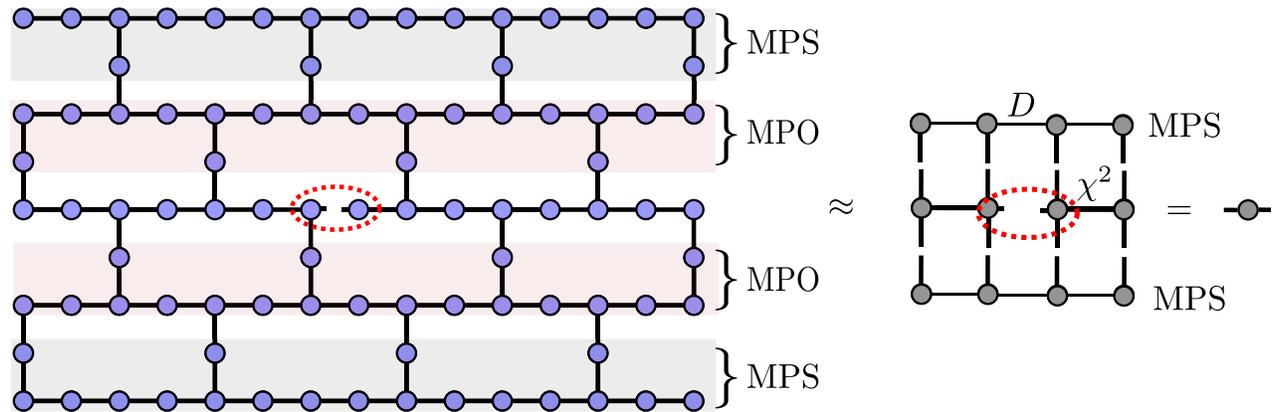


FIG. 10. Approximating the edge environment of an edge e of a TNS on the heavy hex lattice of bond dimension χ using a boundary MPS-style scheme [50]. The contraction of the norm network is done by successively (top to bottom and bottom to top) approximating the contraction of pairs of rows of the heavy hex lattice as MPSs of bond dimension D . The resulting contraction is then reduced to the contraction of a pair of MPSs incident to the given row of the TNS.

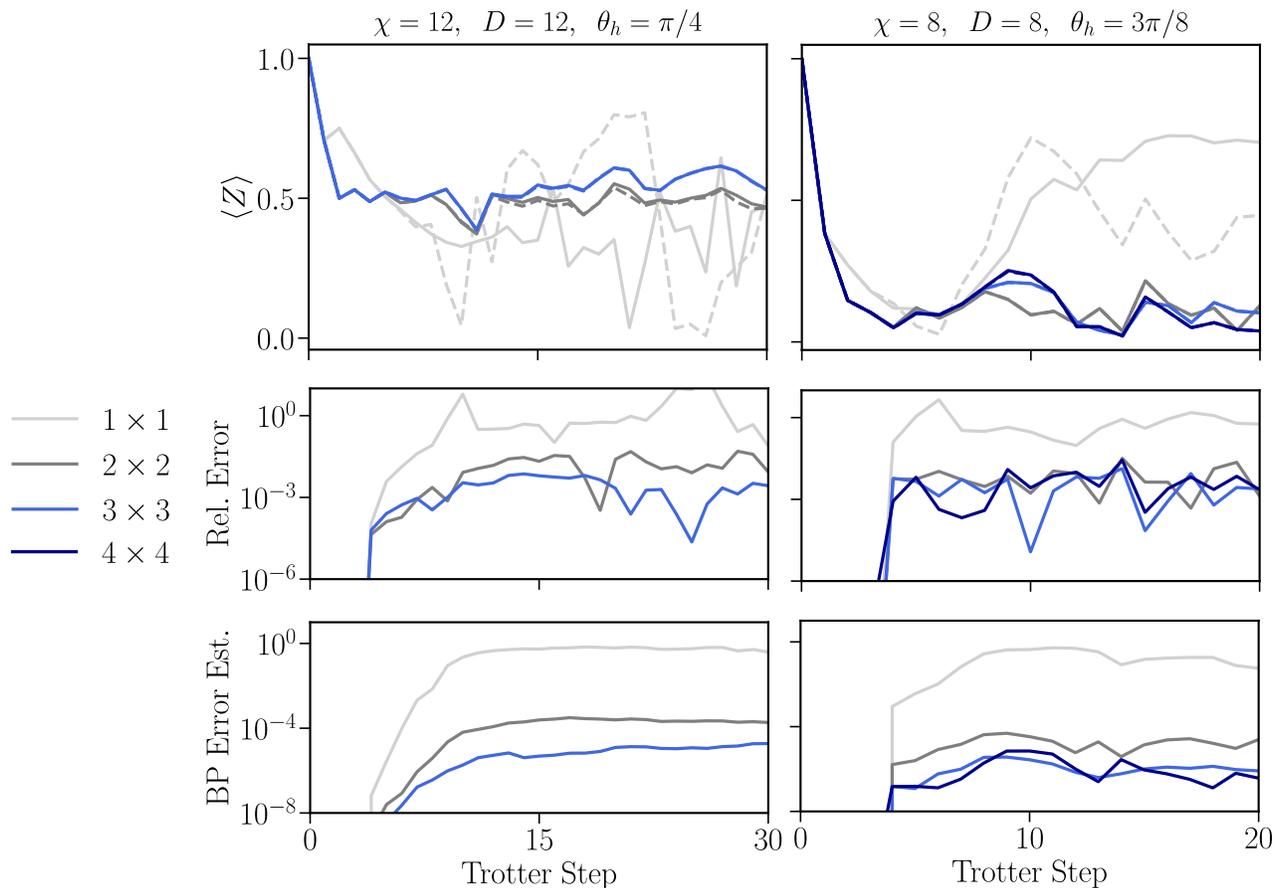


FIG. 11. Dynamics of the single-site magnetization of the kicked Ising model on heavy hex lattices of varying sizes. The TNS is evolved using the BP approximation and the magnetization is approximated with BP (solid lines) and boundary MPS (dashed lines) with the relevant parameters displayed above the plot. The middle plots show the relative error between calculating the expectation values with BP versus boundary MPS. Bottom plots show the BP error estimate based on Eq. (12) for an edge incident to the site where the magnetization is calculated.

calculate is approximately the same as applying the gate exactly and taking the overlap with the truncated state.

APPENDIX C: BOUNDARY MPS FOR THE HEAVY HEX LATTICE

In order to approximate the error of BP using the separability of the edge environment (see Sec. II E) for lattices that are too large to contract exactly, for example, the larger lattices in Fig. 2(d), we employ a method similar to boundary MPS [32,48,49] but generalized to arbitrary network structures [50] to approximately contract the norm network down to a chosen edge to compute the edge environment. Specifically, we approximate the contraction of two rows of the heavy hex as an MPS of fixed bond dimension D —with the external indices of the given region of the TNS being mapped to the dangling indices of the MPS. This allows us to approximate the successive contraction of rows (from top to bottom and bottom to top) of the heavy hex TNS as a sequence of MPS MPO contractions. This process is depicted diagrammatically in Fig. 10.

We also computed the difference between calculating the magnetization of the BP-evolved TNS using the belief-propagation method versus using boundary MPS for the parameters considered in Fig. 2(d). We present the results here in Fig. 11 and observe how close the values obtained are—with the largest lattices showing differences between the two methods, which are on the order of 10^{-4} . The difference between the two methods generally decreases as the system size is increased.

-
- [1] S. Bravyi, D. Gosset, and R. Movassagh, Classical algorithms for quantum mean values, *Nat. Phys.* **17**, 337 (2021).
 - [2] D. S. Wild and A. M. Alhambra, Classical simulation of short-time quantum dynamics, *PRX Quantum* **4**, 020340 (2023).
 - [3] D. Aharonov, X. Gao, Z. Landau, Y. Liu, and U. Vazirani, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023 (Association for Computing Machinery, New York, NY, USA, 2023), p. 945.

- [4] F. Aurate *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature* **574**, 505 (2019).
- [5] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers?, *Phys. Rev. X* **10**, 041038 (2020).
- [6] F. Pan and P. Zhang, Simulation of quantum circuits using the big-batch tensor network method, *Phys. Rev. Lett.* **128**, 030501 (2022).
- [7] T. Ayril, T. Louvet, Y. Zhou, C. Lambert, E. M. Stoudenmire, and X. Waintal, Density-matrix renormalization group algorithm for simulating quantum circuits with a finite fidelity, *PRX Quantum* **4**, 020304 (2023).
- [8] K. Temme, S. Bravyi, and J. M. Gambetta, Error mitigation for short-depth quantum circuits, *Phys. Rev. Lett.* **119**, 180509 (2017).
- [9] E. van den Berg, Z. K. Mineev, A. Kandala, and K. Temme, Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors, *Nat. Phys.* **19**, 1116 (2023).
- [10] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, *Nature* **618**, 500 (2023).
- [11] M. Leifer and D. Poulin, Quantum graphical models and belief propagation, *Ann. Phys. (N. Y.)* **323**, 1899 (2008).
- [12] S. Sahu and B. Swingle, Efficient tensor network simulation of quantum many-body physics on sparse graphs, [arXiv:2206.04701](https://arxiv.org/abs/2206.04701) [quant-ph] (2022).
- [13] C. Guo, D. Poletti, and I. Arad, Block belief propagation algorithm for 2d tensor networks, [arXiv:2301.05844](https://arxiv.org/abs/2301.05844) [quant-ph] (2023).
- [14] J. Tindall and M. Fishman, Gauging tensor networks with belief propagation, [arXiv:2306.17837](https://arxiv.org/abs/2306.17837) [quant-ph] (2023).
- [15] S. Birnkammer, A. Bastianello, and M. Knap, Prethermalization in one-dimensional quantum many-body systems with confinement, *Nat. Commun.* **13**, 7663 (2022).
- [16] M. Kormos, M. Collura, G. Takács, and P. Calabrese, Real-time confinement following a quantum quench to a non-integrable model, *Nat. Phys.* **13**, 246 (2017).
- [17] In terms of absolute difference between our result and the exact value.
- [18] H.-J. Liao, K. Wang, Z.-S. Zhou, P. Zhang, and T. Xiang, Simulation of IBM’s kicked Ising experiment with projected entangled pair operator, [arXiv:2308.03082](https://arxiv.org/abs/2308.03082) [quant-ph] (2023).
- [19] S. Anand, K. Temme, A. Kandala, and M. Zaletel, Classical benchmarking of zero noise extrapolation beyond the exactly-verifiable regime, [arXiv:2306.17839](https://arxiv.org/abs/2306.17839) [quant-ph] (2023).
- [20] T. Begušić and G. K.-L. Chan, Fast classical simulation of evidence for the utility of quantum computing before fault tolerance, [arXiv:2306.16372](https://arxiv.org/abs/2306.16372) [quant-ph] (2023).
- [21] T. Begušić, J. Gray, and G. K.-L. Chan, Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance, [arXiv:2308.05077](https://arxiv.org/abs/2308.05077) [quant-ph] (2023).
- [22] K. Kechedzhi, S. V. Isakov, S. Mandrà, B. Villalonga, X. Mi, S. Boixo, and V. Smelyanskiy, Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments, [arXiv:2306.15970](https://arxiv.org/abs/2306.15970) [quant-ph] (2023).
- [23] P. C. G. Vlaar and P. Corboz, Simulation of three-dimensional quantum systems with projected entangled-pair states, *Phys. Rev. B* **103**, 205137 (2021).
- [24] R. Orús and G. Vidal, Simulation of two-dimensional quantum systems on an infinite lattice revisited: Corner transfer matrix for tensor contraction, *Phys. Rev. B* **80**, 094403 (2009).
- [25] H. N. Phien, J. A. Bengua, H. D. Tuan, P. Corboz, and R. Orús, Infinite projected entangled pair states algorithm improved: Fast full update and gauge fixing, *Phys. Rev. B* **92**, 035142 (2015).
- [26] H. C. Jiang, Z. Y. Weng, and T. Xiang, Accurate determination of tensor network state of quantum lattice models in two dimensions, *Phys. Rev. Lett.* **101**, 090603 (2008).
- [27] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, *Phys. Rev. Lett.* **91**, 147902 (2003).
- [28] S. S. Jahromi and R. Orús, Universal tensor-network algorithm for any infinite lattice, *Phys. Rev. B* **99**, 195105 (2019).
- [29] S.-J. Ran, W. Li, B. Xi, Z. Zhang, and G. Su, Optimized decimation of tensor networks with super-orthogonalization for two-dimensional quantum lattice models, *Phys. Rev. B* **86**, 134429 (2012).
- [30] 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).
- [31] R. Alkabetz and I. Arad, Tensor networks contraction and the belief propagation algorithm, *Phys. Rev. Res.* **3**, 023073 (2021).
- [32] F. Verstraete and J. I. Cirac, *Renormalization algorithms for quantum-many body systems in two and higher dimensions*, Tech. Rep. (2004) [arXiv:cond-mat/0407066](https://arxiv.org/abs/cond-mat/0407066) type: article.
- [33] J. Tindall, A. Searle, A. Alhajri, and D. Jaksch, Quantum physics in connected worlds, *Nat. Commun.* **13**, 7445 (2022).
- [34] A. Searle and J. Tindall, An exact continuous theory for spin systems on graphons, [arXiv:2303.00731](https://arxiv.org/abs/2303.00731) [cond-mat.stat-mech] (2023).
- [35] ITensorNetworks.jl, <https://github.com/mtfishman/ITensorNetworks.jl> (2023).
- [36] L. Wang, I. Pizorn, and F. Verstraete, Monte Carlo simulation with tensor network states, *Phys. Rev. B* **83**, 134421 (2011).
- [37] W. Li, J. von Delft, and T. Xiang, Efficient simulation of infinite tree tensor network states on the Bethe lattice, *Phys. Rev. B* **86**, 195137 (2012).
- [38] R. Alkabetz and I. Arad, Tensor networks contraction and the belief propagation algorithm, *Phys. Rev. Res.* **3**, 023073 (2021).
- [39] J. Pearl, in *Proceedings of the Second AAAI Conference on Artificial Intelligence*, AAAI’82 (AAAI Press, Pittsburgh, Pennsylvania, 1982), p. 133.
- [40] P. Pippian, S. R. White, and H. G. Evertz, Efficient matrix-product state method for periodic boundary conditions, *Phys. Rev. B* **81**, 081103 (2010).
- [41] G. Evenbly, Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops, *Phys. Rev. B* **98**, 085155 (2018).
- [42] D. A. Depireux, J. Z. Simon, D. J. Klein, and S. A. Shamma, Spectro-temporal response field characterization

- with dynamic ripples in ferret primary auditory cortex, *J. Neurophysiol.* **85**, 1220 (2001), PMID: 11247991.
- [43] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, Julia: A fresh approach to numerical computing, *SIAM Review* **59**, 65 (2017).
- [44] M. Fishman, S. R. White, and E. M. Stoudenmire, Codebase release 0.3 for ITensor, *SciPost Phys. Codebases* **4**, 4 (2022).
- [45] GraphTikz.jl, <https://github.com/mtfishman/GraphTikZ.jl> (2023).
- [46] At least for TNS, we believe that BP contraction to compute expectations values is sufficiently accurate for this model based on comparisons to MPS in Fig. 4 and analysis of the validity of the BP approximation based on exact contraction and boundary MPS contraction in Figs. 2 and 11.
- [47] E. G. D. Torre and M. M. Roses, Dissipative mean-field theory of IBM utility experiment, [arXiv:2308.01339](https://arxiv.org/abs/2308.01339) [quant-ph] (2023).
- [48] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Classical simulation of infinite-size quantum lattice systems in two spatial dimensions, *Phys. Rev. Lett.* **101**, 250602 (2008).
- [49] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, Unifying projected entangled pair state contractions, *New J. Phys.* **16**, 033014 (2014).
- [50] L. Ma, M. Fishman, E. M. Stoudenmire, and E. Solomonik, (to be published 2023).