

Supplementary Materials for  
**Fast and converged classical simulations of evidence for the utility of  
quantum computing before fault tolerance**

Tomislav Begušić *et al.*

Corresponding author: Garnet Kin-Lic Chan, gkc1000@gmail.com

*Sci. Adv.* **10**, eadk4321 (2024)  
DOI: 10.1126/sciadv.adk4321

**This PDF file includes:**

Supplementary Text  
Figs. S1 to S10  
Tables S1 to S4

## Supplementary Text

### Detailed descriptions of the lazy TN contractions

We here give a detailed description of the tensor networks formed and contracted in Fig. 2. A generic requirement for us to run lazy BP on a tensor network is that every tensor belongs uniquely to a single site only. The PEPS and PEPO parts of the TN clearly satisfy this by construction, as do the single qubit gates that appear in  $U$ . For the two-qubit gates we perform a SVD, grouping indices spatially, to yield a single tensor per site, connected by a bond. In the case of the  $ZZ$  rotation here, the gate is exactly low-rank with a bond dimension of 2. If we further contract all tensors with the same site,  $j$ , and layer (time-step),  $t$ , (effectively fusing single and two-qubit gates), then we are left with a TN that looks like Fig. 2A, with each unitary layer of gates,  $U_t$ , appearing as a PEPO. This step, which yields a TN with exactly one tensor per site *and* layer,  $T_{j,t}$ , is not strictly necessary but helps in clarity of presentation. We thus initially have

$$\langle 0^{\otimes N} | U^\dagger O U | 0^{\otimes N} \rangle = \sum_{\{x\}} \prod_{j \in 1 \dots N} T_{j,0}^{\psi^\dagger} T_{j,0}^\psi \prod_{j \in 1 \dots N, t \in 1 \dots T} T_{j,t}^{U^\dagger} T_{j,t}^U \prod_{j \in 1 \dots N} T_{j,T+1}^\Phi, \quad (1)$$

where  $T_{j,0}^\psi$  is the tensor of site  $j$  in the PEPS  $\psi$  representing the initial state at  $t = 0$ ,  $T_{j,t}^U$  is the tensor of site  $j$  and layer  $t$  of the unitary  $U$  presented as a PEPO, and  $T_{j,T+1}^\Phi$  is the tensor at site  $j$  of the PEPO representation of initial operator  $\Phi = O$ , and the sum over  $\{x\}$  refers to the sum over all indices appearing on each tensor which for brevity we do not show here. As we evolve the PEPS and PEPO we will generate intermediate tensors  $T_{j,t}^\psi$  and  $T_{j,t}^\Phi$ . Each tensor carries a subset of all indices  $\bar{x}_T$ , either physical (connecting between layers of the same site) or virtual (connecting between sites of the same layer – green bonds in Fig. 2A). The physical indices and the virtual indices of  $U$  here are always of dimension 2 whereas the virtual indices of the  $\psi$  and  $\Phi$  tensors generically (after some evolution) will be between 1 and  $\chi$ .

There are three different tensor networks we run need to run ‘lazy’ BP on. The first is when we evolve the PEPS by one step and want to combine and compress the next layer of gates:

$$U(t+1)|\psi(t)\rangle \rightarrow |\psi(t+1)\rangle. \quad (2)$$

We first form a sub network of the tensors in layers  $\{t, t+1\}$ . Since there are dangling indices (which would connect to layer  $t+2$ ) and we want to compress, we need to run ‘2-norm’ BP. Specifically we form the 2-norm TN:

$$\langle \psi(t) | U^\dagger(t+1) U(t+1) | \psi(t) \rangle = \sum_{\{x: \delta[t+1, t+2]\}} \prod_{j \in 1 \dots N} T_{j,t}^{\psi^\dagger} T_{j,t+1}^{U^\dagger} T_{j,t+1}^U T_{j,t}^\psi, \quad (3)$$

where with the sum and delta we imply that only indices connecting layers  $t+1$  to  $t+2$  are traced over and the remaining indices are summed normally (as shown in Fig. 2B(i)). If we consider site  $j$  and neighbor  $k$  then each of the four tensors at each site is connected by a virtual bond to its matching partner: two of size  $\leq \chi$  and two of size 2. The two BP messages for this bond,  $m_{j \rightarrow k}$ ,  $m_{k \rightarrow j}$ , should thus be of size  $\leq 4\chi^2$ , and we will be able to factorize them as matrices, by fusing all the ‘bra’ ( $\dagger$ ) and all the ‘ket’ indices separately. The expression to compute an updated message (which we initialize as the uniform distribution), is given by:

$$m'_{j \rightarrow k} = \sum_{\{x\} \setminus \{j \rightarrow k\}} T_{j,t}^{\psi^\dagger} T_{j,t+1}^{U^\dagger} T_{j,t+1}^U T_{j,t}^\psi \prod_{l \in \text{neighbors}(j): l \neq k} m_{l \rightarrow j}, \quad (4)$$

where the sum denotes that the indices shared between site  $j$  and  $k$  are not summed over. Performing this message update contraction lazily, i.e. making using of associativity to find an optimized sequence of intermediates, is much cheaper than explicitly forming a single tensor site, which (for our degree 3 lattice) would have size  $64\chi^6$ , whereas this lazy method scales like  $\mathcal{O}(\chi^4)$ . This is the same scaling as Refs. (12, 31).

The second case we run lazy BP on is the PEPO evolution step. Here we want to compress the operator PEPO along with the next layer of gates:

$$U^\dagger(t-1)\Phi(t)U(t-1) \rightarrow \Phi(t-1). \quad (5)$$

Again we have dangling indices (two per site that would connect to layer  $t-2$ ) and thus must form the 2-norm of this operator in order to run BP. Specifically, we form:

$$\langle \langle U^\dagger(t-1)\Phi(t)U(t-1) | U^\dagger(t-1)\Phi(t)U(t-1) \rangle \rangle = \sum_{\{x: \delta[t-1, t-2]\}} \prod_{j \in 1 \dots N} T_{j,t-1}^{U^\dagger} T_{j,t}^{\Phi^\dagger} T_{j,t}^U T_{j,t-1}^{U^\dagger} T_{j,t}^\Phi T_{j,t-1}^U \quad (6)$$

where with the sum and delta we imply that only indices connecting layers  $t - 1$  to  $t - 2$  are traced over and the remaining indices are summed normally (as shown in Fig. 2B(ii)). In this case, considering two neighboring sites  $j$  and  $k$ , we have connecting bonds: two of dimension up to  $\chi$  coming from the central operator PEPOs  $\Phi$  and  $\Phi^\dagger$  and four of dimension 2 coming from the unitary PEPOs. The two BP messages in this case will thus be of size  $16\chi^2$  and will again be able to be factorized as matrices due to the 2-norm structure. The messages are updated according to the contraction:

$$m'_{j \rightarrow k} = \sum_{\{x\} \setminus \{j \rightarrow k\}} T_{j,t-1}^{U\dagger} T_{j,t}^{\Phi\dagger} T_{j,t-1}^U T_{j,t-1}^{U\dagger} T_{j,t}^\Phi T_{j,t-1}^U \prod_{l \in \text{neighbors}(j): l \neq k} m_{l \rightarrow j}, \quad (7)$$

which again we perform as an optimized sequence of pair-wise contractions, to find a scaling of  $\mathcal{O}(\chi^4)$ .

The final TN we run lazy BP on is the operator expectation once the PEPS  $\psi$  and PEPO  $\Phi$  have reached each other at time  $\tau \sim T/2$ . In this case the network  $\langle \psi(\tau) | \Phi(\tau + 1) | \psi(\tau) \rangle$  has no dangling indices and so we do not need to form the 2-norm, instead approximating the contraction directly. This TN takes the form:

$$\langle \psi(\tau) | \Phi(\tau + 1) | \psi(\tau) \rangle = \sum_{\{x\}} \prod_{j \in 1 \dots N} T_{j,\tau}^{\psi\dagger} T_{j,\tau+1}^\Phi T_{j,\tau}^\psi, \quad (8)$$

(as shown in Fig. 2B(iii)). In this case there are three bonds between any two neighbors  $j$  and  $k$ , each of dimension up to  $\chi$ , and thus the two BP messages will have total size  $\chi^3$  and will *not* be factorizable as matrices. The message update is given by:

$$m'_{j \rightarrow k} = \sum_{\{x\} \setminus \{j \rightarrow k\}} T_{j,\tau}^{\psi\dagger} T_{j,\tau+1}^\Phi T_{j,\tau}^\psi \prod_{l \in \text{neighbors}(j): l \neq k} m_{l \rightarrow j}. \quad (9)$$

Here contracting the three tensors at a single site would yield a tensor with size  $\chi^9$ , whereas the the optimized sequence of contractions scales like  $\mathcal{O}(\chi^6)$ . Note this is only for the MIX method where both  $\psi$  and  $\Phi$  have bond dimensions  $\chi$ .

In the current implementation we perform a single iteration of BP by updating the messages in parallel, which is to say, each new message is computed by contraction with messages from the previous round only (see Fig. 2C). We measure the change in a message as:

$$\Delta_{j,k} = |m'_{j,k} - m_{j,k}|_1 \quad (10)$$

and stop iterating when the maximum change,  $\Delta_{j,k}$ , over all messages drops below a specified threshold, here typically taken as  $5 \times 10^{-6}$  for single precision.

### Error analysis

To estimate the convergence error of the MIX method, we propose three independent metrics. First, for every  $\theta_h$  we consider the standard deviation  $\sigma$  of three results with the highest available  $\chi$ . These are presented in Fig. S1. The highest  $\sigma$  is found at  $\theta_h = 10\pi/32$  with the value of  $1.2 \times 10^{-3}$ .

Second, a linear extrapolation using the last three points is compared to the most converged (highest  $\chi$ ) result for each  $\theta_h$  and the error is computed as

$$\Delta = |\langle O \rangle - \langle O \rangle_{\chi \rightarrow \infty}|.$$

Figure S1 shows the data at different  $\theta_h$  and different  $\chi$ , as well as the extrapolation fits and the associated error estimates  $\Delta$ . Although the extrapolation approach estimates larger errors than the standard deviation metric, all error estimates remain well below 0.01.

Third, we recall that the expectation values presented in our results are computed without any renormalization of the wavefunction or operator. Here, we compare these expectation values to their normalized counterparts  $\langle O \rangle/N$ , where  $N$  is the appropriate norm (e.g.,  $N_{\text{MIX}} = N_O N_\psi$  for the mixed Schrödinger-Heisenberg approach). Figure S2 presents these two types of expectation values, along with their average  $\langle O \rangle_{\text{av}} = (\langle O \rangle + \langle O \rangle/N)/2$  for the MIX result of the 20-step expectation value  $\langle Z_{62} \rangle$ . Generally, the normalized and unnormalized methods approach the converged result from above and below, respectively, whereas their average converges faster by cancellation of errors. We can therefore use  $\langle O \rangle_{\text{av}}$  as an estimate of the exact value and  $\bar{\Delta} = |\langle O \rangle - \langle O \rangle_{\text{av}}|$  to estimate the error of the unnormalized result. Again, we find that the errors are conservatively below 0.01, with the highest  $\bar{\Delta} = 0.0023$  for  $\theta_h = 9\pi/32$ .

### 1. *TN timing analysis*

In Fig. S8, Fig. S9, and Fig. S10, we show the total time to run the MIX, PEPS and PEPO TN methods respectively. Note that not all data appearing in the main text was timed or used the same CPU/GPU and thus not all the data appears here. We include in the total time all aspects of the algorithm. After the actual contractions involved in the L2BP compression and L1BP contraction, the second most time consuming part comes from automatic contraction tree optimization. We note that for all the methods and 5 step observables the time is about 10 seconds for a single CPU core, and this is dominated by overhead. It is only the 20 step  $\langle Z_{62} \rangle$  simulations where one can see the scaling with  $\chi$ , and here we show both total time using a single CPU core (of a AMD EPYC 7742) time vs using a single GPU (NVIDIA A100). We show three different angles,  $\theta_h = 6\pi/32, 8\pi/32, 10\pi/32$ . Notably at  $\theta_h = 6\pi/32$  the entanglement growth is slowest and in fact the dynamic singular value cutoff ( $\kappa = 5 \times 10^{-6}$ ) limits the bond dimension of the PEPS or PEPO, whereas at the two higher values the bond dimension becomes limited by the maximum allowed value  $\chi$  and we see the asymptotic complexity regime. Due to the fact that the final TN that we L1BP contract in the MIX method has regions connected by 3 bonds of size up to  $\chi$  (see Fig. 2B(iii)), this method has a higher scaling of  $\chi^6$ , which is evident in Fig. S8E, as compared to the  $\chi^4$  scaling evident in Fig. S9E and Fig. S10E for the PEPS and PEPO methods. It offers better performance despite this since it effectively doubles the accessible evolution time for a given  $\chi$ .

## Supplementary Figures

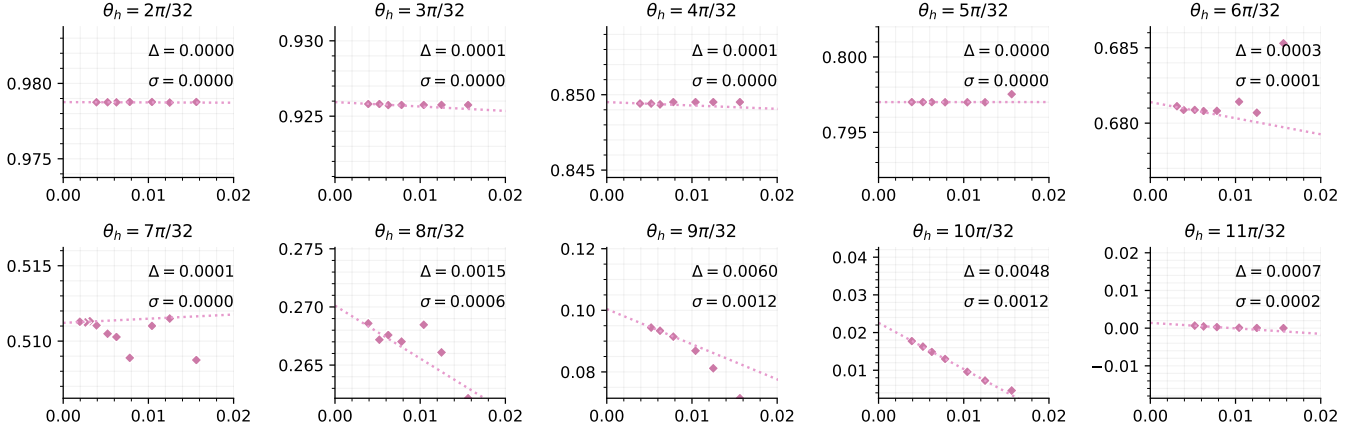


FIG. S1. Convergence of the MIX expectation value  $\langle Z_{62} \rangle$  after 20 steps and linear extrapolation (dashed line) using the last three points, i.e., three points with the highest available  $\chi$ .  $\Delta = |\langle Z_{62} \rangle - \langle Z_{62} \rangle_{\chi \rightarrow \infty}|$  is the difference between the expectation value at the highest  $\chi$  and the extrapolated value, while  $\sigma$  corresponds to the standard deviation of the last three points.

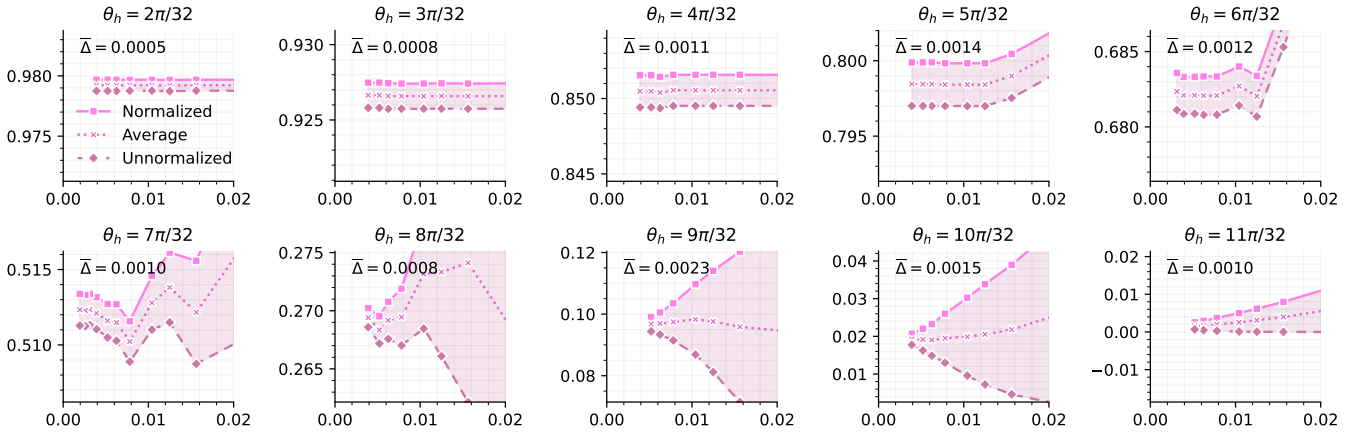


FIG. S2. Convergence of the unnormalized MIX expectation value  $\langle Z_{62} \rangle$  after 20 steps compared with the normalized  $\langle Z_{62} \rangle / N_{\text{MIX}}$  and average  $\langle Z_{62} \rangle_{\text{average}} = (\langle Z_{62} \rangle + \langle Z_{62} \rangle / N_{\text{MIX}}) / 2$  expectation values.  $\bar{\Delta} = |\langle Z_{62} \rangle - \langle Z_{62} \rangle_{\text{average}}|$  for  $\langle Z_{62} \rangle$  evaluated at the highest available  $\chi$  for each  $\theta_h$ . Note for  $\theta_h \leq 6\pi/32$   $\bar{\Delta}$  stops decreasing as the truncation cutoff  $\kappa = 5 \times 10^{-6}$  starts to dominate over the maximum bond dimension  $\chi$ .

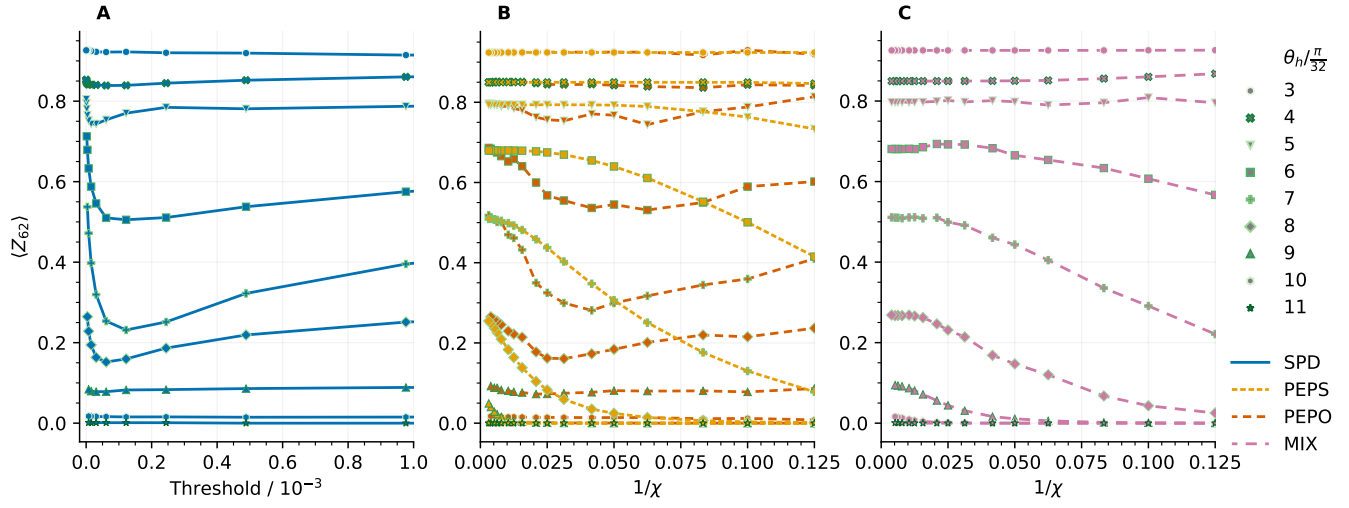


FIG. S3. Same as Fig. 6 but with an extended  $x$ -axis.

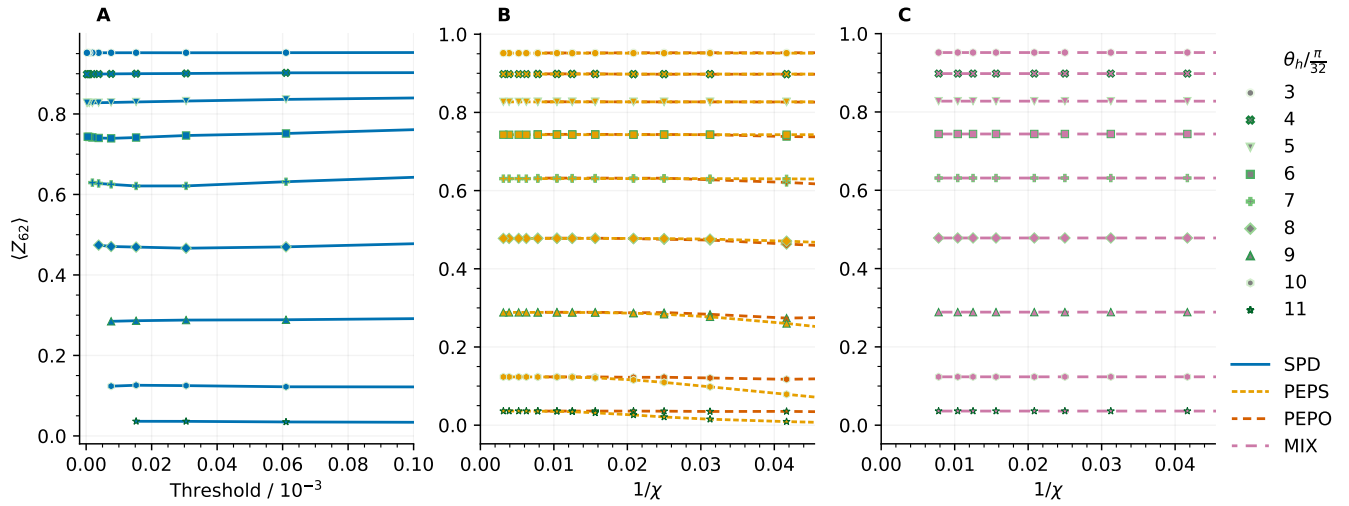


FIG. S4. Same as Fig. S3 but for the 9-step circuit.

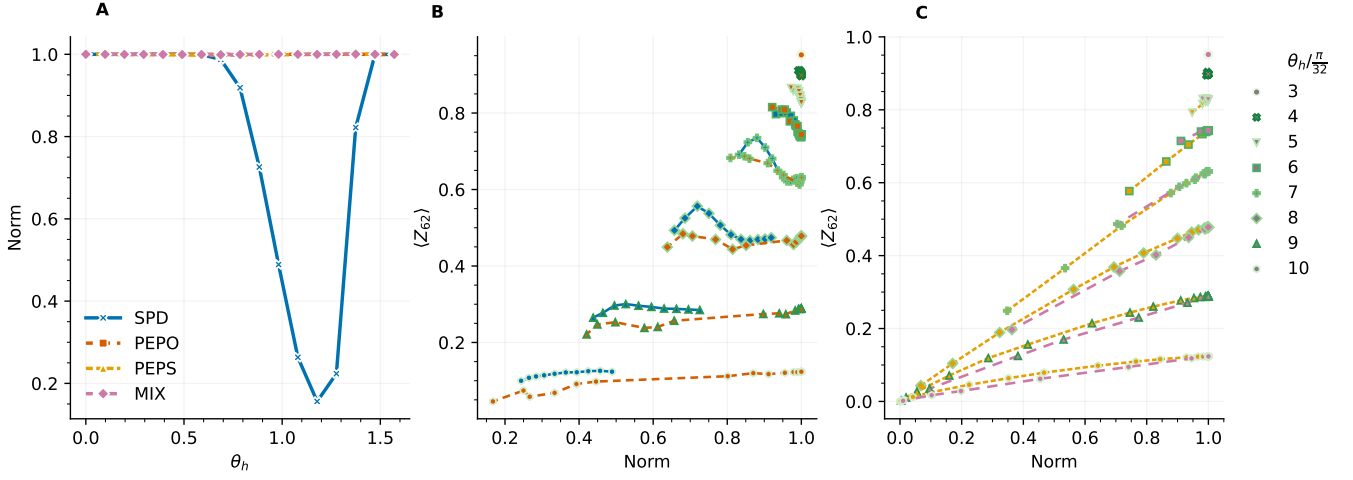


FIG. S5. Same as Fig. 7 of the main text but for the 9-step circuit.

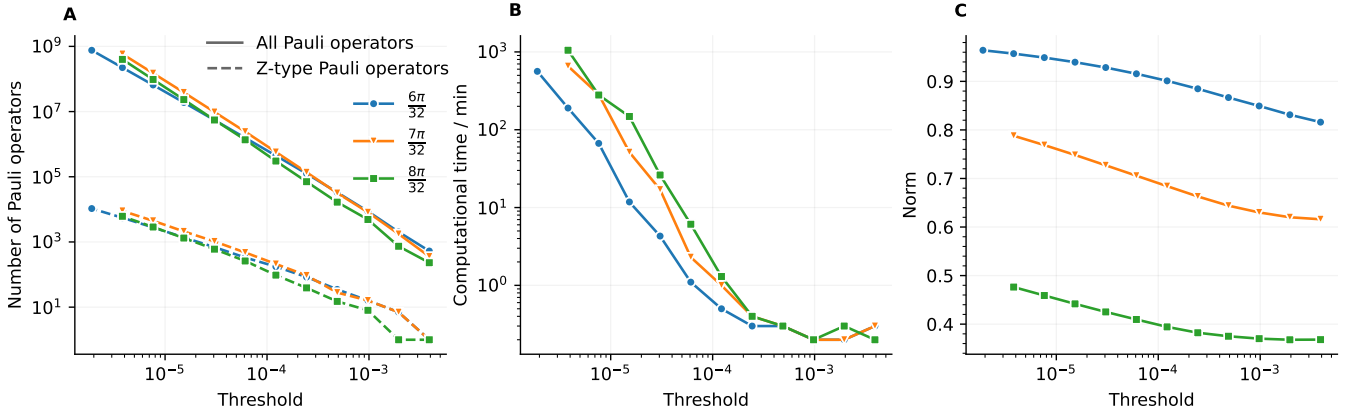


FIG. S6. **Computational cost scaling of SPD.** Total number of Pauli operators (solid, A), the number of Z-type Pauli operators (dashed, A), computational time (B), and the Frobenius norm of the observable (C) at the end of the SPD simulation of  $\langle Z_{62} \rangle$  (20-step quantum circuit) for  $\theta_h = 6\pi/32, 7\pi/32, 8\pi/32$ . The total number of Paulis and computational time scale roughly quadratically with the inverse of the threshold, the number of Z-type Paulis is inversely proportional to the threshold, whereas the norm is approximately logarithmic in this range of thresholds and for  $\theta_h$  values presented.

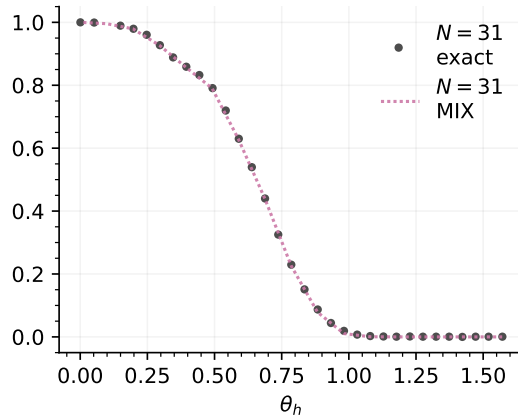


FIG. S7. Exact and MIX simulations of the 31-qubit model (13).

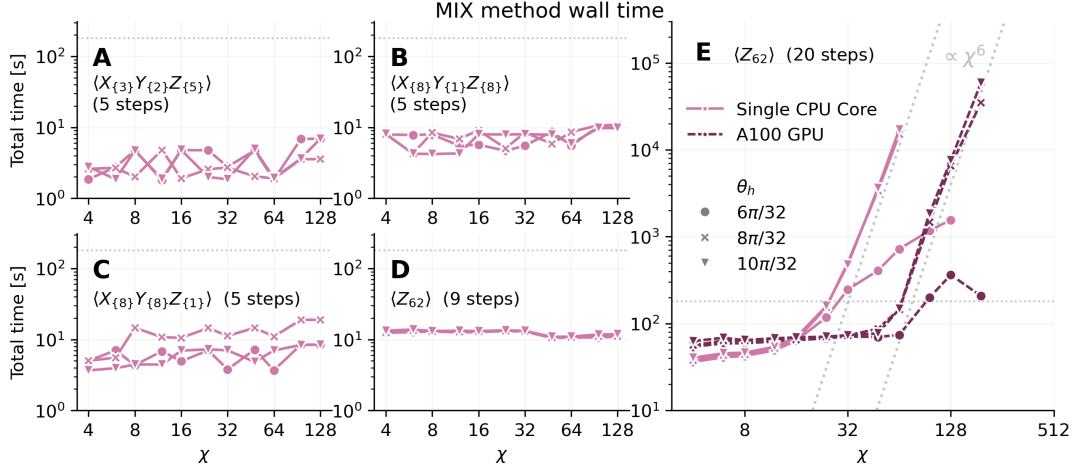


FIG. S8. **Timings for the MIX TN algorithm.** **A-C:** the multi qubit observables at depth 5; **D, E:** the  $\langle Z_{62} \rangle$  observable at depth 9 and 20. The time includes all steps including: circuit construction, L2BP compression, L1BP contraction of both the norms and observable, automatic contraction tree optimization (which can be cached from run to run). The horizontal line denotes 3 minutes, and the diagonal lines in panel **E** show a  $\chi^6$  scaling as a guide. Three different angles,  $\theta_h$ , are shown, and two different contraction backends are shown - either a single core of a AMD EPYC 7742 or a single NVidia A100.

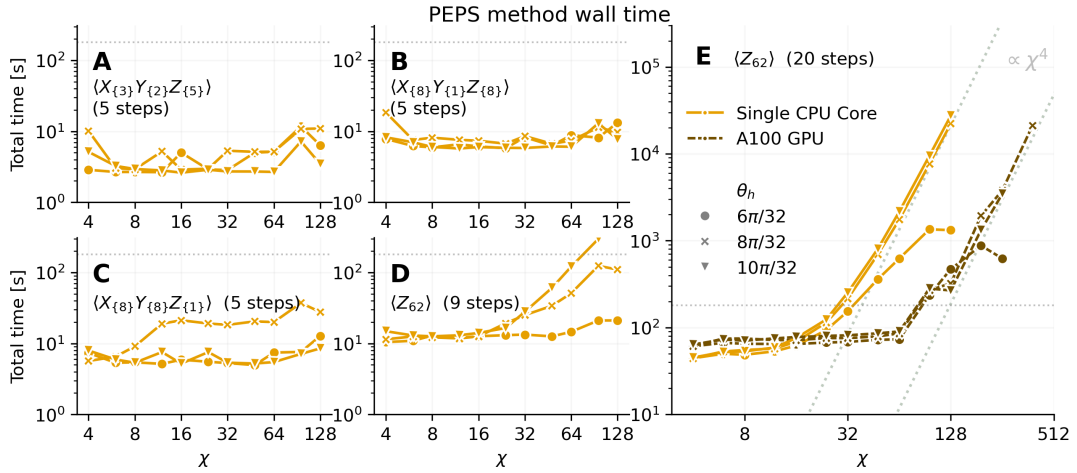


FIG. S9. **Timings for the PEPS TN algorithm.** **A-C:** the multi qubit observables at depth 5; **D, E:** the  $\langle Z_{62} \rangle$  observable at depth 9 and 20. The time includes all steps including: circuit construction, L2BP compression, L1BP contraction of both the norm and observable, automatic contraction tree optimization (which can be cached from run to run). The horizontal line denotes 3 minutes, and the diagonal lines in panel **E** show a  $\chi^4$  scaling as a guide. Three different angles,  $\theta_h$ , are shown, and two different contraction backends are shown - either a single core of a AMD EPYC 7742 or a single NVidia A100.

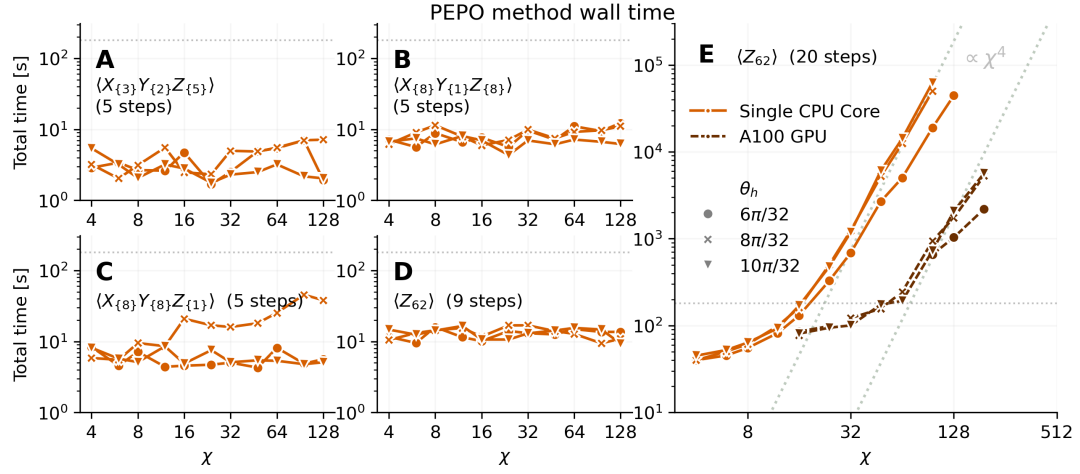


FIG. S10. **Timings for the PEPO TN algorithm.** **A-C:** the multi qubit observables at depth 5; **D, E:** the  $\langle Z_{62} \rangle$  observable at depth 9 and 20. The time includes all steps including: circuit construction, L2BP compression, L1BP contraction of both the norm and observable, automatic contraction tree optimization (which can be cached from run to run). The horizontal line denotes 3 minutes, and the diagonal lines in panel **E** show a  $\chi^4$  scaling as a guide. Three different angles,  $\theta_h$ , are shown, and two different contraction backends are shown - either a single core of a AMD EPYC 7742 or a single NVidia A100.

### Supplementary Tables

TABLE S1. **Thresholds for SPD simulations presented in Figs. 3 and 4.**

	SPD 10 s	SPD
Magnetization $M_Z$	$10^{-3}$	0
$X_{13,29,31}Y_{9,30}Z_{8,12,17,28,32}$	$1.5 \times 10^{-4}$	$5 \times 10^{-6}$
$X_{37,41,52,56,57,58,62,79}Y_{75}Z_{38,40,42,63,72,80,90,91}$	$3.5 \times 10^{-4}$	$10^{-5}$
$X_{37,41,52,56,57,58,62,79}Y_{38,40,42,63,72,80,90,91}Z_{75}$	$3.5 \times 10^{-4}$	$10^{-5}$
$Z_{62}$	$8 \times 10^{-4}$	

TABLE S2. Timings (wall time in minutes using 4 CPU cores) for SPD simulations of high-weight Pauli observables with the thresholds reported in the third column of Table S1.

$\theta_h/(\pi/32)$	$X_{\{3\}}Y_{\{2\}}Z_{\{5\}}$	$X_{\{8\}}Y_{\{1\}}Z_{\{8\}}$	$X_{\{8\}}Y_{\{8\}}Z_{\{1\}}$
1	0.4	0.3	0.3
2	0.4	0.6	1.4
3	0.4	3.0	12.6
4	0.8	10.5	40.9
5	1.7	36.8	54.4
6	5.0	78.0	154.3
7	12.4	141.0	212.0
8	20.3	263.2	362.6
9	44.8	217.1	356.1
10	31.7	294.7	258.1
11	21.2	215.1	277.4
12	7.7	108.1	198.1
13	1.5	20.4	114.6
14	0.4	3.0	14.8
15	0.3	0.3	0.4

TABLE S3. Thresholds, numbers of Pauli operators at the end of simulation, and wall time (on 6 CPU cores) for the 9-step and 20-step SPD simulations of Fig. 5. Note that the number of Pauli operators operated with during the simulation can be much greater, especially in the examples where few Paulis are left at the end of the simulation.

$\theta_h/(\pi/32)$	9 steps			20 steps		
	Threshold	$N_{\text{Pauli}}$	Timing / min	Threshold	$N_{\text{Pauli}}$	Timing / min
1	$2^{-22}$	1,723	0.5	$2^{-22}$	10,527	0.4
2	$2^{-22}$	50,391	2.0	$2^{-22}$	705,896	0.4
3	$2^{-22}$	1,037,984	0.4	$2^{-22}$	16,447,691	7.4
4	$2^{-22}$	15,630,592	1.8	$2^{-22}$	215,196,358	190.9
5	$2^{-22}$	226,944,006	44.2	$2^{-21}$	885,442,824	803.5
6	$2^{-21}$	999,386,176	229.8	$2^{-19}$	760,142,525	562.7
7	$2^{-19}$	833,860,251	197.5	$2^{-18}$	582,723,097	656.2
8	$2^{-18}$	865,920,748	489.2	$2^{-18}$	398,851,143	1050.2
9	$2^{-17}$	357,673,648	216.9	$2^{-17}$	22,894,192	861.9
10	$2^{-17}$	285,852,981	288.7	$2^{-17}$	2,372,374	820.2
11	$2^{-16}$	33,062,799	117.0	$2^{-17}$	88,810	799.5
12	$2^{-16}$	16,659,256	120.2	$2^{-16}$	172	181.4
13	$2^{-16}$	54,559,484	121.0	$2^{-16}$	0	144.3
14	$2^{-17}$	580,390,756	261.4	$2^{-16}$	0	294.7
15	$2^{-20}$	375,828,476	64.4	$2^{-16}$	73,030,357	635.5

TABLE S4. **Maximum bond dimension  $\chi$  used for the TN simulations presented in the main text.** Note for 9 steps, the norm reaches  $\sim 1$  for all  $\theta_h$  (see Fig. S5A), implying sufficient  $\chi$ , at 256, 64, and 16 for PEPS, PEPO and MIX methods respectively. The dynamic singular value truncation threshold in all cases was set to  $\kappa = 5 \times 10^{-6}$ . This, combined with the different entanglement growth at different  $\theta_h$ , is what leads to slightly different accessible  $\chi$  values for each method at 20 steps. The ‘Mix 3min’ method shown in Fig. 3 was executed on a NVidia 4070 Ti.

Method	5 steps	9 steps	20 steps ( $\theta_h \leq 8\pi/32$ )	20 steps ( $\theta_h \geq 9\pi/32$ )
PEPS	64	320	320	320
PEPO	64	128	320	256
MIX	64	128	256	192
MIX 3min	64	n/a	64	64