nature portfolio

https://doi.org/10.1038/s41586-024-07998-6

Supplementary information

Phase transitions in random circuit sampling

In the format provided by the authors and unedited

Supplement to Phase transition in Random Circuit Sampling

Google Quantum AI and Collaborators

CONTENTS

Appendix A: General RCS with XEB theory

We show in this appendix that, under quite general conditions (see Eq. (A10)), the effect of noise in XEB can be approximated as a global depolarizing channel. We use this to write an XEB estimator from any smooth and O(1) function $f(p_j)$ of the ideal probabilities p_j.

It is non-trivial but true that the density of probabilities from a Haar random pure quantum state is uniform in the probability simplex [1–3]

$$dP(p_1, \ldots, p_D) = (N-1)!\, dp_1 \cdots dp_D \ , \tag{A1}$$

where $D = 2^n$ for n qubits. The corresponding marginal distribution for any one probability p_j is the Porter-Thomas (exponential or beta) distribution [4]. That is, for all j we have

$$dP(p_j) = (D-1)(1-p_j)^{D-2} dp_j \tag{A2}$$

$$\to D e^{-Dp_j} dp_j \ . \tag{A3}$$

In the previous expression the bitstring index j is fixed, and the distribution is over quantum states sampled uniformly in Hilbert space (Haar measure).

One can sample a vector of the probabilities corresponding to a Haar random pure quantum state by sampling D probabilities according to the distribution of Eq. (A3), and then normalizing the result so that $\sum_j p_j = 1$ [1, 2, 5]. Note that the sum of the independent p_j is already $\sum_j p_j = 1 + \mathrm{O}(1/\sqrt{D})$ before normalization. That is, for large D the normalization introduces a small correlation between the previously independent p_j that can be typically ignored.

Approximate sampling of a random quantum circuit can be described by the probabilities

$$p_j^F = F p_j + (1-F)\Xi_j \tag{A4}$$

where F corresponds to the fidelity, p_j is the ideal or simulated probability for the jth bitstring output of the quantum circuit, and Ξ_j is a function over bitstrings corresponding to the effect of noise. In the quantum case, ρ is the output of an experiment, $p_j^F = \langle j| \rho |j\rangle$, $F = \langle \psi| \rho |\psi\rangle$ where $|\psi\rangle$ is the ideal noiseless output, and Ξ is defined by the equation $\rho = F|\psi\rangle\langle\psi| + (1-F)\Xi$. Note that $\sum_j \Xi_j = 1$. For simplicity we sometimes denote $\Xi_j = 1/D$, the global depolarizing channel.

In cross-entropy benchmarking (XEB) we use the expectation value of a random variable $f(p_j)$, which is defined as a function of the ideal probabilities p_j. That is, we associate the real value $f(p_j)$ to each sampled bitstring $|j\rangle$. We require $f(p_j)$ to be O(1) and f smooth. For linear XEB $f(p_j) = Dp_j - 1$ and for log XEB $f(p_j) = \log(Dp_j) +$ Euler constant [6]. In the following we assume that the output distribution is sufficiently close to the Porter-Thomas distribution, see Refs. [4, 6] and below.

The expectation value of $f(p_j)$ when sampling with noisy probabilities p_j^F is

$$\sum_j p_j^F f(p_j) = F \sum_j p_j f(p_j) + (1-F) \sum_j \Xi_j f(p_j) \ .$$

The sum on the left hand side is an expectation value estimated with RCS sampling, within a statistical error $O(1/\sqrt{k})$ where k is the size of the sample. We explain below how to obtain the value of the two sums in the right hand size analytically for large circuits. Therefore solving for F we obtain an estimator of the fidelity for any function $f(p_j)$ as specified above.

Consider first the term $p_j f(p_j)$. The expectation value over random circuits for fixed j is

$$\langle\langle p_j f(p_j)\rangle\rangle = D \int_0^\infty dp\, e^{-Dp}\, p f(p) \ , \tag{A5}$$

where $\langle\langle\cdot\rangle\rangle$ is the average over random circuits. Note that from the assumptions on f above it also follows that $\langle\langle p_j f(p_j)\rangle\rangle$ is $\mathrm{O}(1/D)$. Furthermore, the variance over random circuits for fixed j is

$$\mathrm{Var}(pf(p)) \in \mathrm{O}\left(\frac{1}{D^2}\right) \ . \tag{A6}$$

We saw above that the probabilities p_j are almost independent. Treating the sum over j as a sum of independent and identically distributed (i.i.d.) random variables, we have, by the central limit theorem,

$$\sum_j p_j f(p_j) = D^2 \int_0^\infty dp\, e^{-Dp}\, p f(p) + \mathrm{O}\left(\frac{1}{\sqrt{D}}\right) \tag{A7}$$

Now we consider the term $\Xi_j f(p_j)$. We assume that, when averaged over random circuits for fixed j, the random variables Ξ_j and $f(p_j)$ are independent. Therefore

$$\langle\langle \Xi_j f(p_j)\rangle\rangle = \langle\langle \Xi_j\rangle\rangle \, \langle\langle f(p)\rangle\rangle \ , \tag{A8}$$

where

$$\langle\langle f(p)\rangle\rangle = D \int_0^\infty dp\, e^{-Dp}\, f(p) \ . \tag{A9}$$

We also assume that

$$\langle\langle \Xi_j\rangle\rangle \in \mathrm{O}\left(\frac{1}{D}\right) . \tag{A10}$$

Therefore

$$\mathrm{Var}(\Xi_j f(p_j)) \in \mathrm{O}\left(\frac{1}{D^2}\right) \ . \tag{A11}$$

Treating again the sum over j as a sum of i.i.d. random variables we obtain

$$\sum_j \Xi_j f(p_j) = \langle\langle f(p)\rangle\rangle \sum_j \langle\langle \Xi_j\rangle\rangle + \mathrm{O}\left(\frac{1}{\sqrt{D}}\right) \tag{A12}$$

$$= \langle\langle f(p)\rangle\rangle + \mathrm{O}\left(\frac{1}{\sqrt{D}}\right) \ , \tag{A13}$$

where we used $\sum_j \Xi_j = 1$. We conclude that the averaged effect of noise Ξ_j can be approximated as a totally depolarizing channel.

For linear XEB we have $f(p) = Dp - 1$ and therefore

$$\sum_j p_j f(p_j) = D^2 \int_0^\infty dp\, e^{-Dp}\, p(Dp-1) + \mathrm{O}\left(\frac{1}{\sqrt{D}}\right) = 1 + \mathrm{O}\left(\frac{1}{\sqrt{D}}\right) \tag{A14}$$

$$\sum_j \Xi_j f(p_j) = D \int_0^\infty dp\, e^{-Dp}\, (Dp-1) + \mathrm{O}\left(\frac{1}{\sqrt{D}}\right) = 0 + \mathrm{O}\left(\frac{1}{\sqrt{D}}\right). \tag{A15}$$

We obtain the same result for log XEB $f(p_j) = \log(Dp_j) + \text{Euler constant}$ [6]. Therefore we have

$$F \simeq \langle Dp - 1 \rangle_{\text{experiment}} \tag{A16}$$

$$\simeq \langle \log(Dp_j) + \text{Euler constant} \rangle_{\text{experiment}}\ . \tag{A17}$$

We now check numerically at what depth the output distribution becomes Porter-Thomas. Figure 1 shows that the probabilities p_j truly follow a Porter-Thomas distribution (as measured by the Kolmogorov-Smirnov test) only if the linear XEB is exponentially close to its limit value. The scaling in the x axis comes from the variance (see also Ref. [6])

$$\mathrm{Var}\,(\text{lin XEB}) \simeq D^2\, \mathrm{Var}\left(p_j^2\right) = \frac{2}{D}. \tag{A18}$$

Nevertheless, we find numerically and experimentally that XEB serves as an estimator of fidelity before this point, and closer to the transition point in Fig. 1a of the main text, as we don't require exponential $O(1/\sqrt{D})$ precision for this estimation.

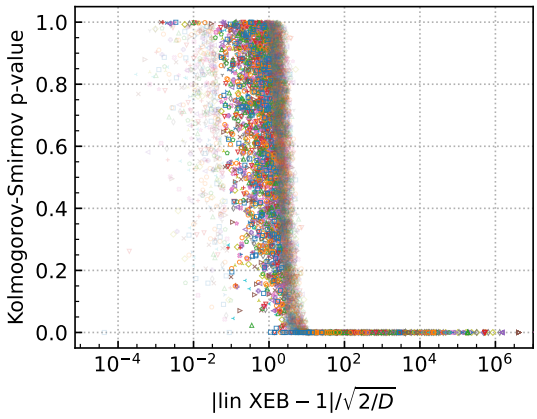


FIG. 1. The y-axis is the Kolmogorov-Smirnov $p-$value between the probabilities p_j at a given depth and the Porter-Thomas distribution. The x axis is the distance of linear XEB to the ideal value in units of standard deviation. Each point corresponds to a different circuit size (with number of qubits ranging from $n = 8$ to $n = 25$) for a given fixed depth. Lighter points correspond to datapoints outside the 90% two-sided confidence interval. For all the instances, the pattern ABCDCDAB is used.

Appendix B: Device characterization and benchmarking

1. Gate Optimization

The gate fidelities of the quantum processor are carefully optimized through a series of steps. The first step involves shaping of the flux pulses used to realize the iSWAP-like gates, schematically shown in Fig. 2A. Here the computational states of two qubits, $|10\rangle$ and $|01\rangle$, are brought into resonance by pulsing the qubit frequencies ω_1 and ω_2 to nearly identical values. An inter-qubit coupling g is then pulsed to a maximum value of $g_{\max} \sim -13$ MHz over $t_\mathrm{p} = 20$ ns to enable a complete population transfer from $|10\rangle$ to $|01\rangle$.

An important error channel for such a two-qubit gate is the off-resonant oscillation between the $|11\rangle$ and $|02\rangle$ (as well as $|20\rangle$) states, which may result in appreciable leakage outside the computational space at the end of the pulses. One possible strategy for mitigating leakage is through simultaneous optimization of $g_{\max}$ and t_p such that the minima in leakage and iSWAP angle errors are synchronized [7]. However, due to the spread in qubit anharmonicities, such an optimization needs to be done for each individual qubit pair and is therefore a time-consuming process. An alternative method is to increase the rise time of the coupler pulse such that the transitions $|11\rangle \leftrightarrow |02\rangle$ and $|11\rangle \leftrightarrow |20\rangle$ are both adiabatic, thereby eliminating the need for synchronization. The leakage rates per iSWAP gate r_l, measured using a method adapted from Floquet calibration and applied to the two-exitation subspace [8], are shown in the bottom panel of Fig. 2A. We observe that for short rise times in the coupler pulse ($t_\mathrm{r} = 0$ ns), r_l in excess of 10^{-3} is observed for most qubit pairs. The leakage rate is suppressed as t_r is increased to 2 ns, although outlier qubit pairs with $r_l > 10^{-3}$ are still observed. For $t_\mathrm{r} = 4$ ns, all pairs tested show $r_l < 4 \times 10^{-4}$. We therefore employ $t_\mathrm{r} = 4$ ns for experiments described in this work.

The pulse shape optimization of the iSWAP-like gates has led to a reduction in two-qubit cycle Pauli errors r_p in parallel two-qubit XEB from an initial median value of 1.01×10^{-2} to 8.4×10^{-3}, as shown in Fig. 2B. In the same plot, we show two additional optimization steps that have further improved gate fidelities: By optimizing qubit frequency placements on the 2D grid [9] to mitigate cross-talk and coupling to two-level system (TLS) defects, we reduce r_p to 6.5×10^{-3}. Finally, r_p is reduced to only 5.9×10^{-3} by shortening the execution time for the single-qubit gates from 25 ns to 18 ns.

After minimizing the cycle errors in two-qubit parallel XEB experiments, we benchmark performance of larger

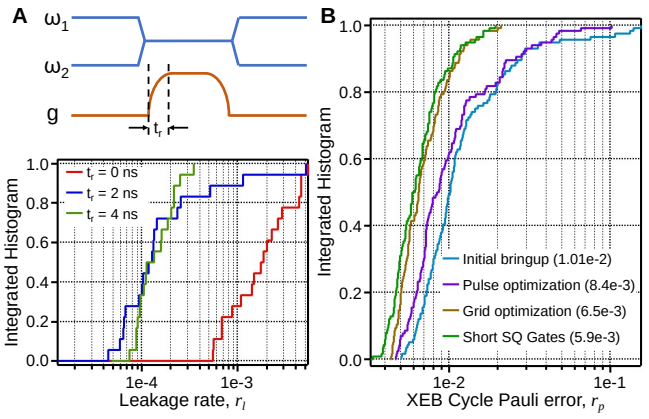


FIG. 2. Gate fidelity optimizations. (A) Upper panel: Schematic showing the flux pulses which detune the qubit frequencies (ω_1 and ω_2) and the inter-qubit coupling g during the iSWAP-like gate. A cosine filter with a rise time t_r is applied to the pulse on g. Lower panel: Integrated histograms of leakage per iSWAP-like gate, measured with three different values of t_r. Each histogram includes an identical set of 19 qubit pairs. (B) Integrated histograms of two-qubit Pauli error per cycle (which includes contributions from two single-qubit gates and one iSWAP-like gate) obtained from parallel XEB taken after different optimization steps indicated by the legend. Each histogram includes all qubit pairs on the quantum device. The median values of different histograms are quoted within the parentheses of the legend.

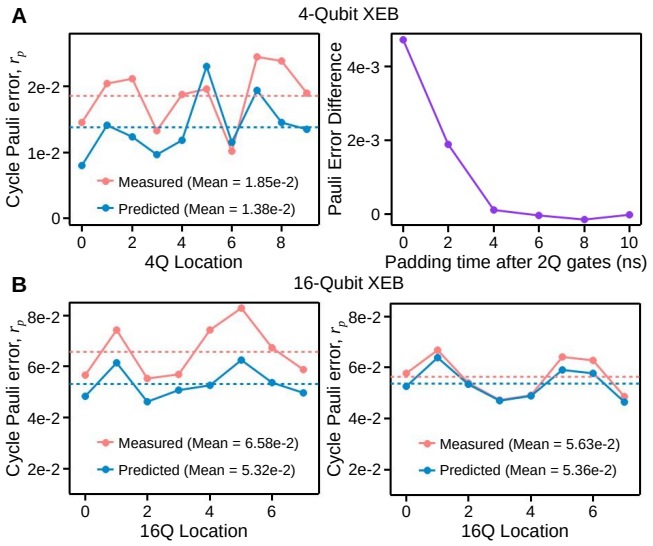


FIG. 3. Mitigating impact of z-tails. (A) Left panel: Comparison between the cycle Pauli error of a 4-qubit XEB experiment and the prediction from parallel 2-qubit XEB experiments. Horizontal axis corresponds to different 4-qubit choices. Dashed line indicates the mean values of the measured and predicted errors. Right panel: Mean difference between the measured and predicted 4Q XEB cycle errors as a function of padding times after the iSWAP-like gates. (B) Left panel: Comparison between the cycle Pauli error of a 16-qubit XEB experiment and the prediction from parallel 2-qubit XEB experiments. Horizontal axis corresponds to different 16-qubit choices. Dashed line indicates the mean values of the measured and predicted errors. Right panel: Same as the left panel but with median qubit detunings during the iSWAP-like gate reduced from 80 MHz to 40 MHz.

system sizes by performing a 4-qubit XEB experiment on ten different choices of 4 qubits across the quantum processor. We detect a substantial difference between the measured four-qubit cycle error and the predicted four-qubit cycle error based on two-qubit XEB measurements, as shown in the left panel of Fig. 3A. The average 4-qubit cycle errors are over 30% higher than predicted values. Through further characterizations, this discrepancy is understood to be arising from distortions in qubit flux pulses which lead to a slow settling of the qubit frequencies even after the pulses have nominally ended (a.k.a. "z-tails"). To mitigate the impact of z-tails, we pad the moments between the two-qubit gates and single-qubit gates in the random circuits by an idling time. The right panel of Fig. 3A shows the average difference between the measured and predicted four-qubit XEB cycle errors as a function of the padding time, where we observe that a padding time of 4 ns is sufficient to reduce the difference to nearly 0. This additional padding time has been applied to experiments described in this work.

Having reached agreements between two-qubit and four-qubit XEB experiments, we compare two-qubit parallel XEB predictions with 16-qubit XEB experiments. The initial result is shown in Fig. 3B, where we again find that the measured 16-qubit XEB cycle errors are 24% higher than predictions, even with padding between single- and two-qubit gates. To reduce this discrepancy, we have re-optimized the qubit frequency placements and reduced the detunings of the qubits during the iSWAP-like gates by a factor of two. The 16-qubit parallel XEB cycle errors measured after this qubit frequency re-optimization agrees closely with the predicted values from two-qubit parallel XEB measurements.

2. Benchmarking of gates and readout

In order to construct the error model for a random circuit, we use several experiments to predict the error rate of each element. The single qubit error is calibrated through Randomized Benchmarking using only $\pi/2$-pulses Fig. 5A. For the RB, we use 5 different depths logarithmically spaced up to a thousand Clifford, with 10 random Clifford circuit instances with 600 repetition per number of cycles and circuit. The two-qubit dressed error is measured through parallel XEB after optimizing for a phased-fSim model Fig. 5C. We used 20 random circuit instances, with 10 linearly spaced depths up to 150 cycles. The readout error is measured by preparing a random bitstring state and measuring the probability of wrong labeling of each qubit Fig. 5B. The error is

averaged between the measurement error of the state $|0\rangle$ and $|1\rangle$. Finally, T_1 and T_2 echo, used for the idling on the edges of each patches, is measured through a standard population decay experiment and echo measurement Fig. 5D and E. respectively. Finally Fig. 5F shows the improvement over the results from [6] with the dashed line reporting the average fidelity achieved at the time. Every aspect of the experiment has improved, with a notable contribution from readout fidelity.

In Fig. 6, we report the angles of the iSWAP-like gate measured with parallel XEB. We note that the c-phase of the gate is now closer to $\pi/10$ compared to $\pi/6$ in [6].

Appendix C: Additional experimental data

1. RCS experiment on a 70 qubits device: SYC-70

In this section, we present the RCS experiment on a 70-qubits device characterized in Fig. 4. Figure 7 shows experiments without phase matching (similar to [6]) and with phase matching (similar to what is presented in the main text of this manuscript). In the Table I of the main text we report the estimated fidelity for the experiment without phase matching.

When performing a two-qubit gate, the actual unitary applied to the qubits differs from the ideal fSim by extra single-qubit Z-rotations from two sources: 1) the qubits are detuned during the gate, and 2) the qubit interaction Hamiltonian terms are not time-independent but rather oscillate due to the frequency difference between the qubits. The rotations arising from (1) do not depend on the time when the gate is applied, but the rotations arising from (2) do depend on this time, with a time-dependent phase $\gamma(t) = 2\pi(f_1 - f_0)t$.

When running quantum circuits, we typically implement Z-rotations as "virtual" gates by changing the phase of applied microwave pulses. This is equivalent to a circuit-level transformation where Z gates are pushed through the circuit by commuting them past other gates. The extra Z rotations associated with fSim gates can also be handled in this way by compiling them into the pulse sequence; in this case we say the fSim gates are "phase-matched". We can also ignore these extra Z rotations when compiling and then account for them in simulation by applying the appropriate time-dependent unitary for each gate instead of the ideal fSim unitary; in this case we say the fSim gate is not phase-matched. Note that Z-rotations only commute through fSim when θ is 0 or $\pi/2$, that is, for c-phase-like or iSWAP-like gates. If the gate is not exactly iSWAP-like, then commuting Z-rotations through it for phase-matching introduces some error, which we can see in the slightly lower XEB fidelity when using phase-matched gates.

2. Adding noise

In order to probe the noise induced phase transition, we artificially increase the single qubit error rate by adding random rotations after each layer of single qubit gates in the circuit run on the hardware. The random single qubit gates are of the form:

$$U = Z^z Z^a X^x Z^{-a} \tag{C1}$$

where z and x are sampled from a normal distribution centered on zero and with a standard deviation given by the injected noise amplitude. The axis a is randomly sampled from a normal distribution centered on -1 with a standard deviation of 1. In order to avoid correlated noise, the random gates are different from layer to layer in a single circuit and from circuit to circuit. These extra gates are not used in the classical simulation. Figure 8 A shows the insertion of the random single qubit gates is done on each single qubit layer of a random circuit. In Figure 8 B we verify that adding these extra single qubit gates results in an average noise that scales as the square of the error angle, as expected.

3. Noise phase transition extended data

In this appendix, we show the full dataset used for the characterization of the noise induced phase transitions identified in the main text. See Figs. 9 and 10.

4. Weak-link model with local noise

We performed an experiment corroborating the behaviour of the weak-link model under local noise, see main text and Sec. D 3. We rewrite Eq. (2) in main text splitting the contribution of the left and right fidelities as

$$\text{linear XEB} = \lambda^{d/T} F_{\text{left}}^d + \lambda^{d/T} F_{\text{right}}^d + (F_{\text{left}} F_{\text{right}})^d \,. \tag{C2}$$

We increases the noise only on the right side of the chain. We see that for low noise the last term dominates and the linear XEB decreases proportionally to the added error. However, for sufficiently large noise in the right side, the linear XEB becomes independent on the added noise. The reason is that the two last terms in Eq. (C2) become negligible compared to the first term.

We probe this behavior experimentally on a chain of 20 qubits with a weak-link applied with a period of $T = 8$. In Fig. 11 we indeed observe that initially the fidelity decays linearly with the added noise. For very strong noise however the linear XEB plateaus at some value, indicating an insensitivity to added noise on the right side.

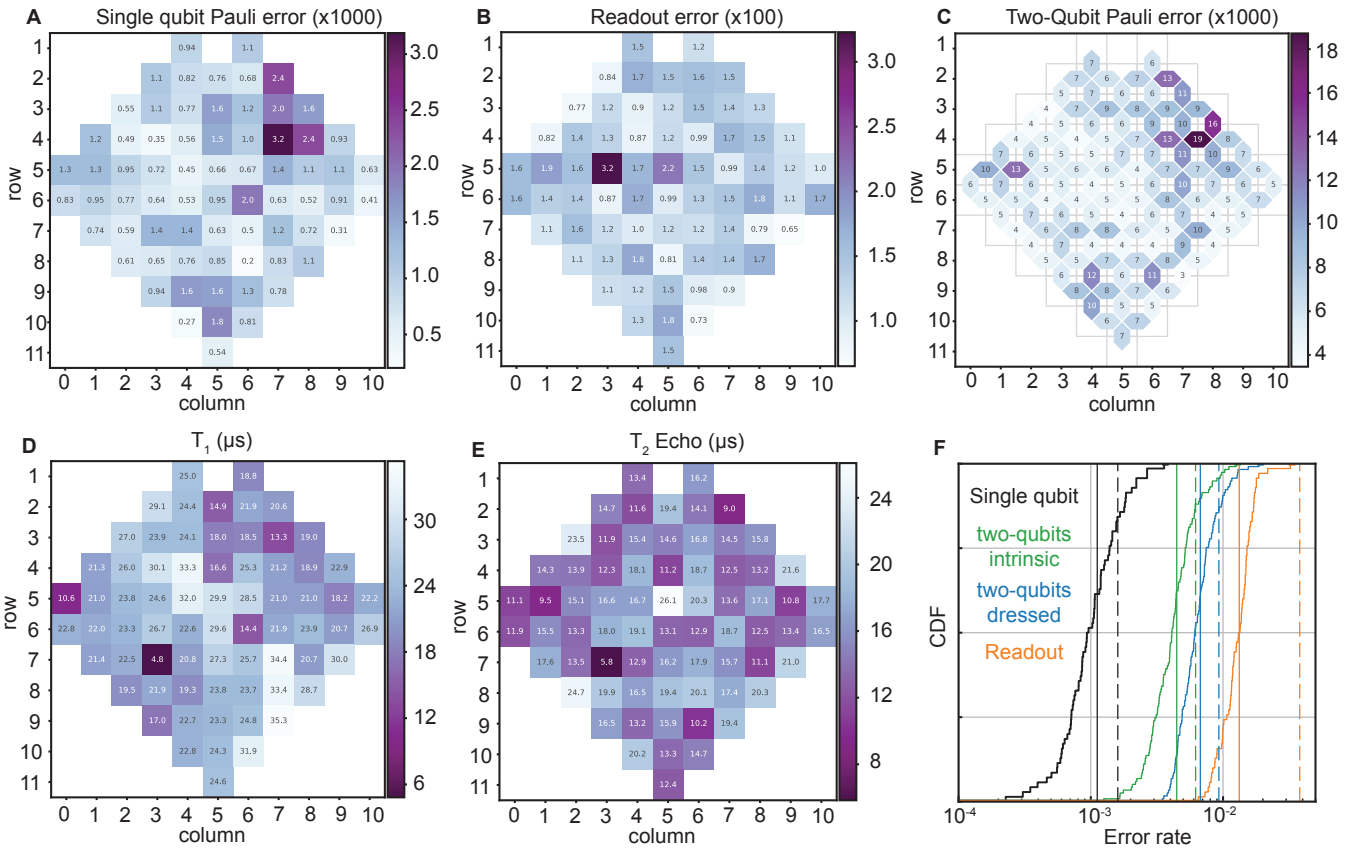


FIG. 4. **Benchmarking of the device for SYC-70**: Benchmarking of the random circuits elements. **A**: Single qubit Pauli error rate measured with Randomized Benchmarking. **B**: Readout error rate measured by preparing random bitstrings and averaging the errors over the bitstrings. **C**: Two qubit Pauli error rate measured with parallel 2-qubit XEB. **D** and **E**: T_1 and T_2 echo times. **F** CDF of the different errors, continuous vertical line is the average, and dashed line corresponds to the average from Ref. [6].

Appendix D: Linear XEB via population dynamics

1. Population dynamics for the uniformly random single qubit gate ensemble

The linear XEB over circuits may be written as

$$\mathrm{XEB}(d) = 2^n C - 1, \tag{D1}$$

$$C = \sum_z \langle z| U\rho_0 U^\dagger |z\rangle \langle z| \mathcal{E}\left[U\rho_0 U^\dagger\right] |z\rangle, \tag{D2}$$

where $\mathcal{E}$ corresponds to a noisy evolution channel. We now explain how its average can be calculated via population dynamics [10–14].

Consider first a noise free evolution. Note that the average probability has the form of an out-of-time ordered correlator, $C = \sum_z \mathrm{Tr}\{\mathcal{O}_z\rho_0(d)\mathcal{O}_z\rho_0(d)\}$ where $\mathcal{O}_z = |z\rangle\langle z|$, and $|z\rangle = \otimes_{i=1}^n |z_i\rangle, z_i = \{0,1\}$ is an n qubit computational basis state. It can be described in terms of two copies of the evolution $\rho_0(d)\otimes\rho_0(d)$. After averaging over uniformly random (Haar) single qubit gates the dynamics in such doubled operator space is fully described in terms of two invariants: identity operator $\mathbb{1}$ and $\mathcal{B} = (1/3)\sum_{\alpha=x,y,z}\sigma^\alpha\otimes\sigma^\alpha$, where σ^α are Pauli operators.

The average dynamics of a pair of identical operators $\overline{\mathcal{O}(d)\otimes\mathcal{O}(d)}$ in the n qubit system subject to a circuit consisting of cycles with two-qubit gates can be described by a time dependent distribution $P(\{v_i\},d)$ over an n bit register $\{v_i\}, v_i \in \{0,1\}$ corresponding to $\{\mathbb{1}_i, \mathcal{B}_i\}$, respectively. That is,

$$\overline{\mathcal{O}(d)\otimes\mathcal{O}(d)} = \sum_{\{v_i\}} P(\{v_i\},d) \bigotimes_i \left((1-v_i)\mathbb{1}_i + \mathcal{B}_i v_i\right). \tag{D3}$$

For operators which satisfy $\mathcal{O}^2 = 1$ (true for Pauli operators) the coefficients are normalized probabilities $\sum_{\{v_i\}} P(\{v_i\},d) = 1$. Each two-qubit gate defines a Markov process with the update matrix,

$$P(\{v_i\}, d+1) = \sum_{v'_j v'_k} \Omega_{v_j v_k, v'_j v'_k} P(\{v'_i\}, d). \tag{D4}$$

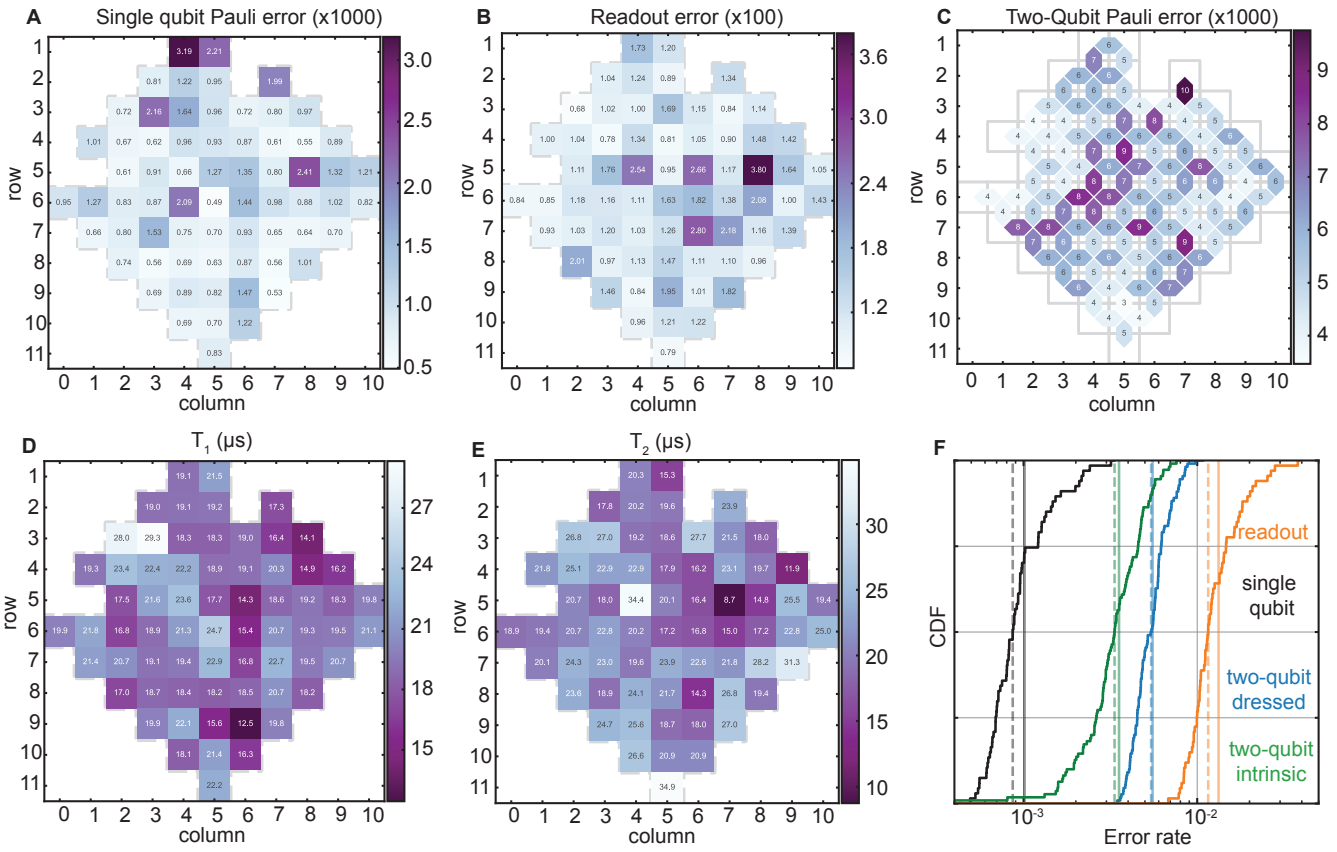


FIG. 5. **Benchmarking of the device for SYC-67**: Benchmarking of the random circuits elements. **A**: Single qubit Pauli error rate measured with Randomized Benchmarking. **B**: Readout error rate measured by preparing random bitstrings and averaging the errors over the bitstrings. **C**: Two qubit Pauli error rate measured with parallel 2-qubit XEB. **D** and **E**: T_1 and T_2 echo times. **F** CDF of the different errors, continuous vertical line is the average, and dashed line corresponds to median.

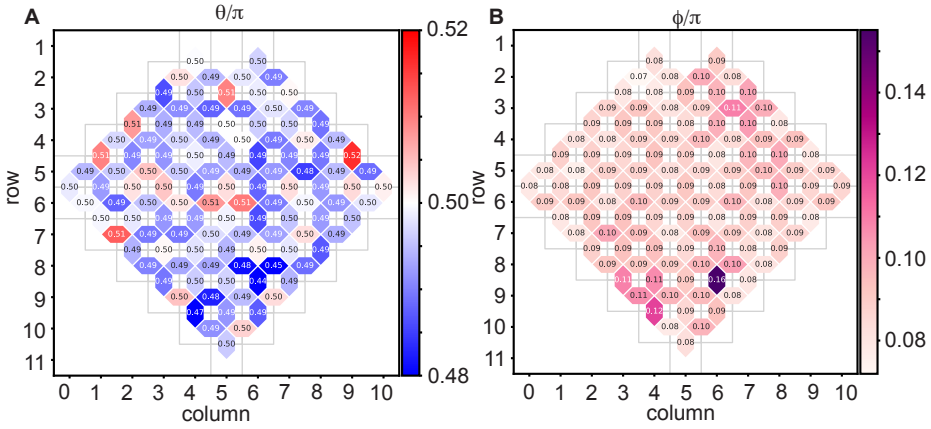


FIG. 6. **iSWAP-like characterization**: Measured angles of the iSWAP-like gate. On average, the angles are $\theta = 0.495(0.009) \times \pi$ and $\phi = 0.09(0.01) \times \pi$

where the indexes j and k correspond to the qubits involved in the corresponding two-qubit gate.

We can take the two-qubit gate to be approximately equal to an iSWAP, $U_{ij} = \exp\left(-i\frac{\pi}{4}(X_iX_j + Y_iY_j)\right)$, for which the population dynamics update corresponds to

$$\hat{\Omega}^{(i,j)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{2}{3} \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{2}{3} & \frac{5}{9} \end{pmatrix}. \tag{D5}$$

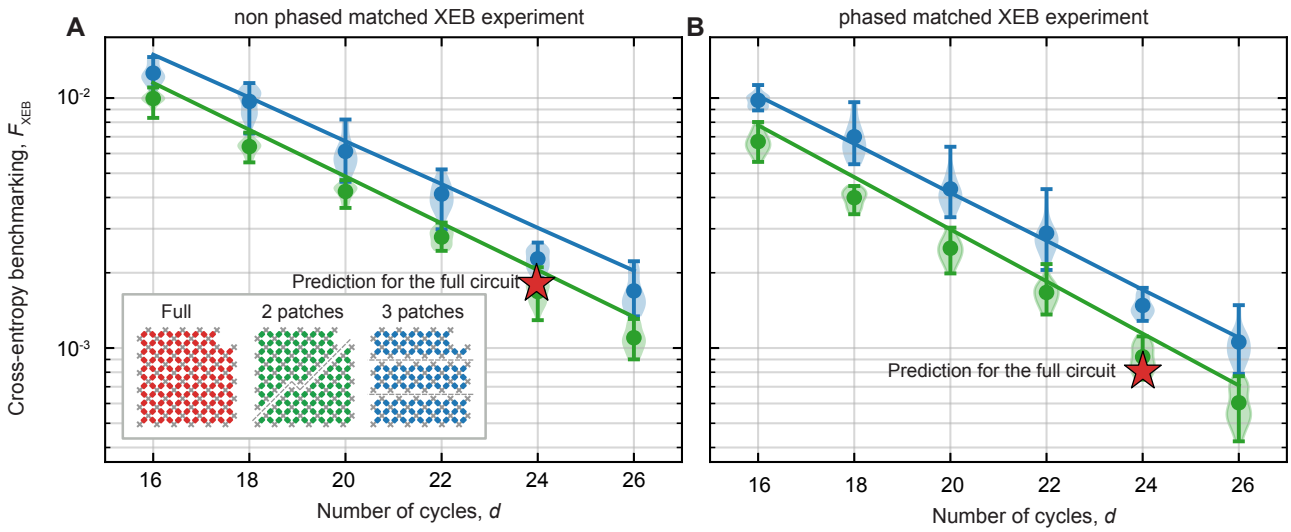


FIG. 7. **Random Circuit Sampling experiment SYC-70**: On a device with 70 qubits, we performed RCS experiments on 3 and 2 patches. Line indicate the predictions from the average error rate obtained by parallel XEB. **A** shows the data without phase matching (see text for more details) and **B** shows data with phase matching, similar to the SYC-67 experiment presented in the main text. The difference between the two-patch and the three-patch fidelities is explained by the larger error rate of the two-qubit gates compared to the idling of the qubits for which two-qubit gates have been removed.

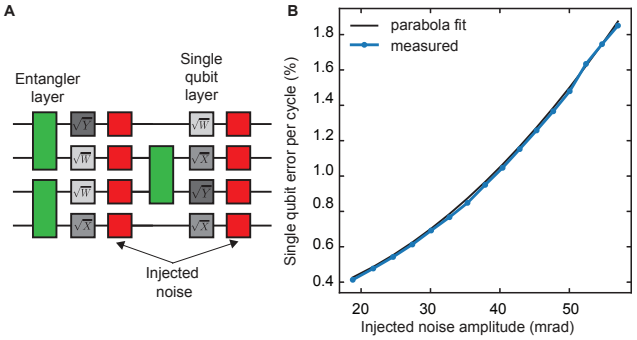


FIG. 8. **Controlling error rate.** **A**: Single qubit gates with random rotations are added after each single qubit layer. **B**: Single qubit error rate with the added random rotations measured with single qubit XEB. The error rate follows a parabolic law.

Elements of the matrix $\hat{\Omega}$ correspond to the transition probabilities between different configurations induced by the application of a two-qubit gate. $\hat{\Omega}_{10,01}$ corresponds to $\mathcal{B}$ hopping from one qubit to another ($\Omega_{10,01} = \frac{1}{3}$ for iSWAP) whereas $\Omega_{10,11}$ corresponds to creation of a new $\mathcal{B}$ ($\Omega_{10,11} = \frac{2}{3}$ for iSWAP).

The contribution of each configuration to XEB is determined by individual invariants, $(\mathbb{1}_i, \mathcal{B}_i) \to (1, 1/3)$ as

$$\mathrm{XEB} = 2^n \sum_{\{v_i\}} \frac{1}{3^{\sum v_i}} P(\{v_i\}, d) - 1. \qquad \text{(D6)}$$

To include the effects of noise the two-qubit gate update rules need to be supplemented with the noise-induced decay rules at each two qubit gate [13],

$$\mathbb{1}_i \mathbb{1}_j \to \mathbb{1}_i \mathbb{1}_j, \qquad \text{(D7)}$$

$$\mathcal{B}_i \mathbb{1}_j \to \exp(-\frac{16}{15} p_2) \mathcal{B}_i \mathbb{1}_j, \qquad \text{(D8)}$$

$$\mathcal{B}_i \mathcal{B}_j \to \exp(-\frac{16}{15} p_2) \mathcal{B}_i \mathcal{B}_j, \qquad \text{(D9)}$$

where p_2 is the two-qubit depolarizing error.

2. Convergence of population dynamics to Porter-Thomas

The initial state for population dynamics is obtained by averaging the initial bitstring $\rho_0 = \prod_i (\mathbb{1}_i + Z_i)/2$ over the first layer of single qubit gates. The result of this averaging is $\prod_i (\mathbb{1}_i + \mathcal{B}_i)/4$. It can be interpreted as equal weight distribution $P(\{v_i\}, 0) = 1/2^n$ over all configurations $\{v_i\}$. After the first layer of one qubit gates XEB $= (4/3)^n$.

In a multi-qubit system a layer of gates corresponds to the evolution under $\hat{\Omega}^{(i,j)}$ applied to each pair of qubits subject to a gate of the layer. The circuit can be characterized by a transfer matrix $\hat{\mathcal{T}}$ that consists of a product of the layers that appears periodically, such that the whole circuit corresponds to $\hat{\mathcal{T}}^d$.

There are two steady states of this Markov chain: (i) the vacuum $\{v_i = 0\}$ for all i, (ii) the thermal state that corresponds to $P(\{v_i\}) = \prod_i p(v_i)$, where $p(0) = 1/4$ and $p(1) = 3/4$. At long times in the noise free Porter-Thomas limit, $C = 2/(2^n + 1)$, and XEB ≈ 1. Note that

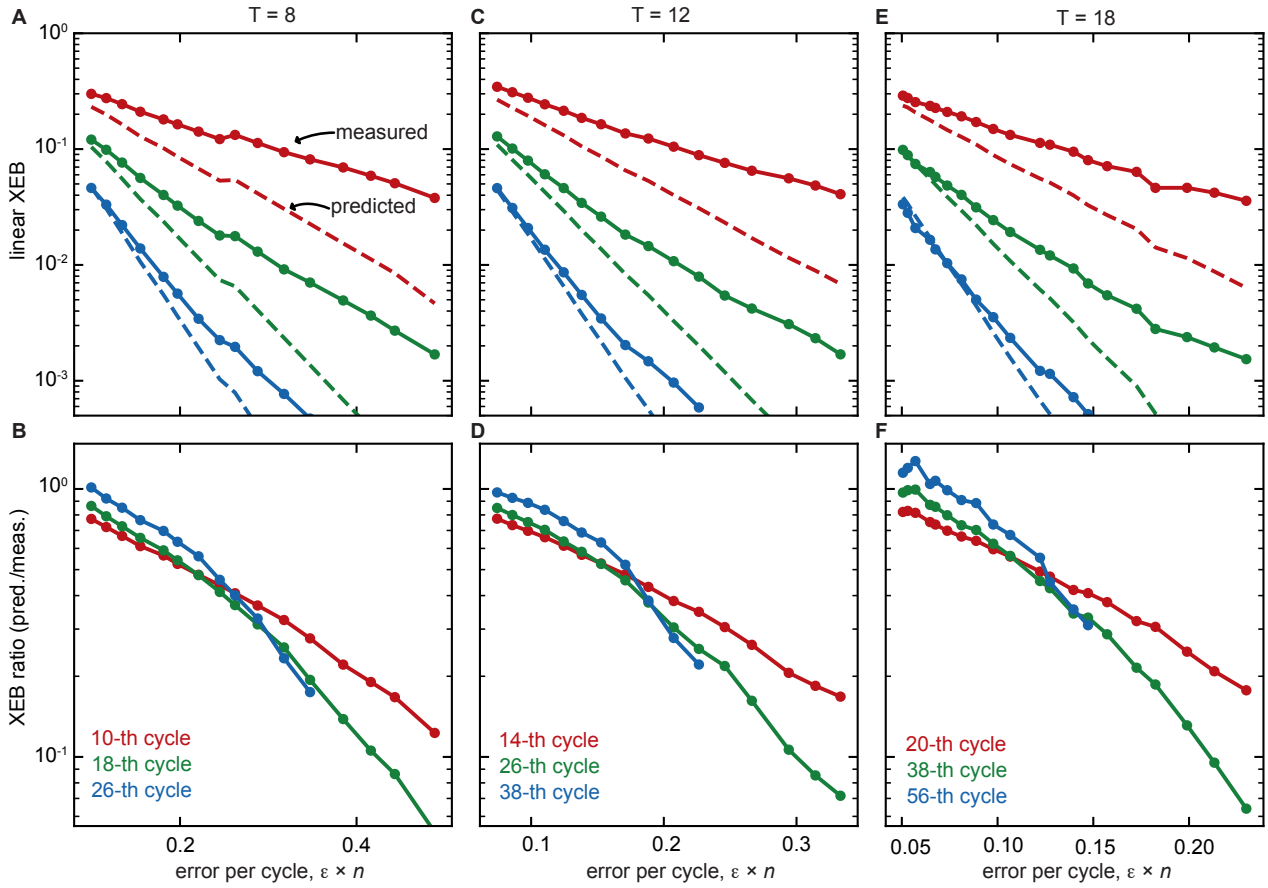


FIG. 9. **Weak-link model**: See main text Fig. 3 for more details. The first row shows the measured XEB value as a function of the error per cycle. In the strong noise regime, the measured XEB value is far from the expected value, whereas in the weak noise regime, and sufficient depth, the measured value is correctly predicted by the component fidelity of the circuit. The second row shows the XEB ratio.

the vacuum configuration $\mathbb{1}^{\otimes n}$ does not evolve, and in the presence of noise in the long depth limit the vacuum is the only remaining configuration. This produces the only non-vanishing contribution to C, resulting in XEB=0.

3. Weak-link model analytical solution

In this section we provide details justifying Eq. (2) of the main text. We consider an example that can be analyzed analytically: a chain with a weak link connecting its two halves A and B, that was introduced in the main text. At the weak link a two-qubit gate is applied only every T cycles. We describe the dynamics of XEB using the population dynamics formalism introduced in Ref. [13].

We assume T is long enough to establish the "thermal" (or Porter-Thomas) state in each half of the chain independently. We introduce probabilities of four possible population dynamics configurations after time T: $g_{00}, g_{01}, g_{10}, g_{11}$ corresponding to both halves in the vacuum state, one half in the vacuum state and one in the thermal state and both halves in the thermal state. Initially all four configurations g_{ij} give order one contributions to linear XEB, despite having exponentially different probability in the $\{v_i\}$ basis, due to the term $1/3^{\sum_{v_i}}$ in Eq. D6.

A single application of the weak link gate after T cycles updates these probabilities as follows,

$$g_{00}(d+T) = g_{00}(d)\ , \tag{D10}$$

$$g_{01}(d+T) = F^{T/2}\frac{1}{4}g_{01}(d)\ , \tag{D11}$$

$$g_{10}(d+T) = F^{T/2}\frac{1}{4}g_{10}(d)\ , \tag{D12}$$

$$g_{11}(d+T) \simeq F^{T}g_{11}(d)\ , \tag{D13}$$

where as before F is the fidelity per layer for the whole chain excluding the weak link, and the factor $1/4$ comes from the two-qubit iSWAP gate. In the last equation we drop the contributions of g_{10} and g_{01} to g_{11} because it adds only an exponentially small contribution to XEB. This is because the initial thermal + vacuum state has ex-

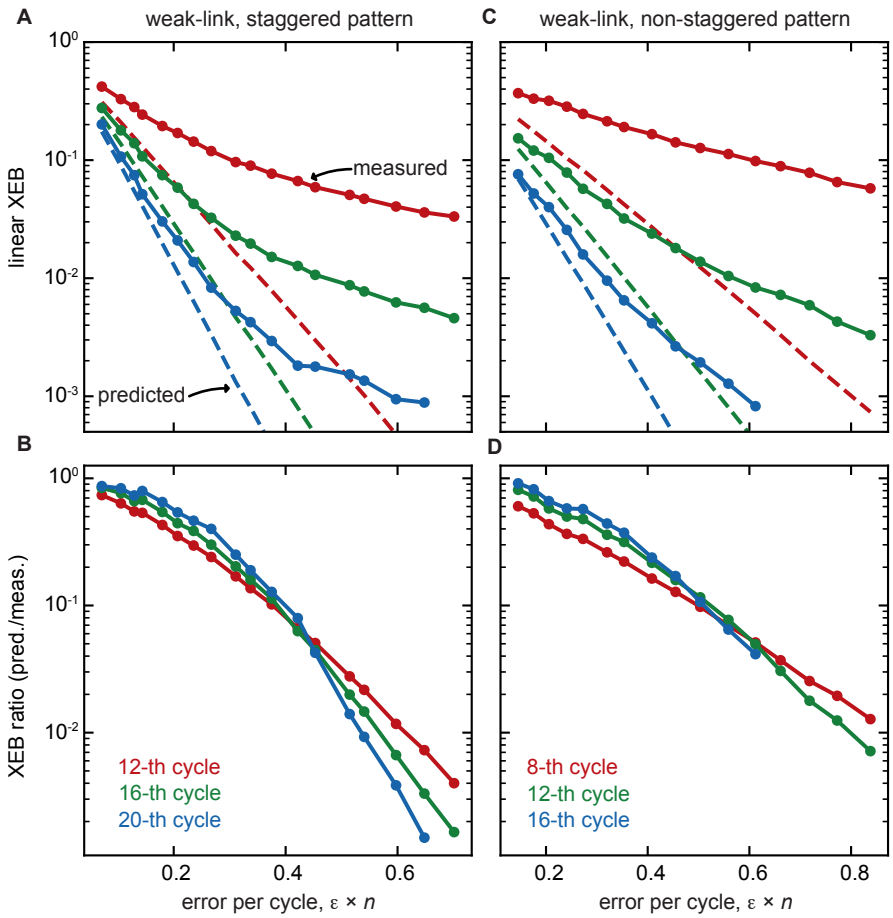


FIG. 10. **Noise phase transition in 2D**: The first row shows the measured XEB as a function of the error per cycle. In the strong noise regime, the measured XEB value is far from the expected value. See main text Fig. 3 for more details about the different patterns. For very strong noise, and large number of cycles the linear XEB value starts to be hard to measure and requires a large number of repetitions, making these measurements challenging.

ponentially smaller probability, as explained above. We therefore find

$$\mathrm{XEB}(mT) = F^{mT} + 2\left(\frac{1}{4}F^{T/2}\right)^m. \tag{D14}$$

This gives a criteria for XEB to serve as a good fidelity estimate for the chain with weak link,

$$F^T > \frac{1}{16}. \tag{D15}$$

4. Numerical analysis of the phase transitions

In this section we provide numerical simulations of XEB dynamics justifying the analysis of the data in the main text. Linear XEB is calculated numerically using the exact mapping on population dynamics introduced above, see Eq. (D6). The time dependence of weights $P(\{v_i\}, d)$ is computed by applying the transfer matrices corresponding to each two-qubit gate, Eq. (D4). This method requires memory that scales exponentially as 2^n because we evolve the full probability vector. At the same time it predicts the dynamics of the average linear XEB in the presence of noise and is quadratically more efficient than direct simulation of a noisy density matrix. Without loss of generality we simulate a simplified model of the noise including only single qubit noise applied to each qubit after each layer of two qubit gates.

We first demonstrate the finite size critical scaling near the dynamical transition. Fig. 12 is a numerical analog of Fig. 2a of the main text, showing the depth dependence of linear XEB for a fixed error rate per qubit $\epsilon = 0.01$ and different size chains. The scaling of linear XEB with the system size changes from growth to decay at the transition point, whose value is approximately size independent. See Sec. E for an analytical description of this phase transition.

The noise induced phase transition (NIPT) is characterized by the change in the depth dependence of the or-

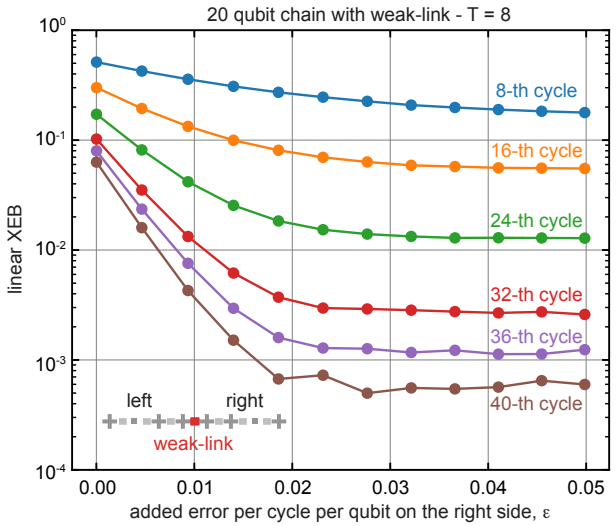


FIG. 11. **Weak-link chain linear XEB with local noise**: We add extra noise only on one side of the weak-link chain. The linear XEB first decays following this added error. However, for strong enough noise, the linear XEB saturates for all depths. For this experiment we have taken a single dataset without extra error, and added coherent errors in the simulation of the circuit for the linear XEB calculation.

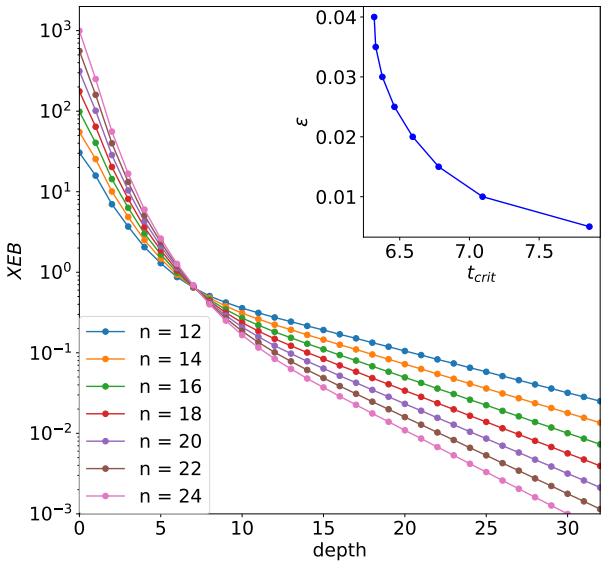


FIG. 12. XEB as a function of depth for different size n qubit chains. The error per qubit per unit time is $\epsilon = 0.01$. Inset shows the dependence of the critical depth on the error per qubit per unit time.

der parameter $\Theta \equiv \exp(-\epsilon n d)/\mathrm{XEB}$. Fig. 13 shows the depth dependence of Θ for different values of the error per cycle $0 \leq \epsilon n \leq 1.34$. At low error the order parameter converges to a constant, whereas in the presence of a sufficiently large error per cycle the order parameter converges to zero. See also Sec. E for an analytical description of this phase transition.

In the case of the weak link model introduced in the main text and in Sec. D 3, where the system is split into two parts with the gates entangling the two parts applied sufficiently rarely, there is a less data intensive procedure to identify the NIPT. This relies on the crossing point of the order parameter Θ as a function of ϵn for different depths of the circuit (separated by a circuit period), as shown in Fig. 14. This procedure was used to identify the transition experimentally in Fig. 3 of the main text, and works well for weak link frequency $1/T < 1$ in 1D. In the absence of the weak link time dependence of the order parameter the method illustrated in Fig. 13 was used to identify the transition point. Numerical simulations for 2D circuits give similar results for the order parameter. The extracted transition points are summarized in the phase diagram presented in Fig. 3 G of the main text.

We also compare the transition point in linear XEB for a uniformly random ensemble of single qubit gates introduced above to the discrete single qubit gate set used in the experiment. The latter maps onto population dynamics in a space of three states per qubit and is more costly to implement, requiring memory 3^n. Fig. 15 shows a comparison of the NIPT location for 16 qubit systems of two different geometries: a chain and a 4×4 system.

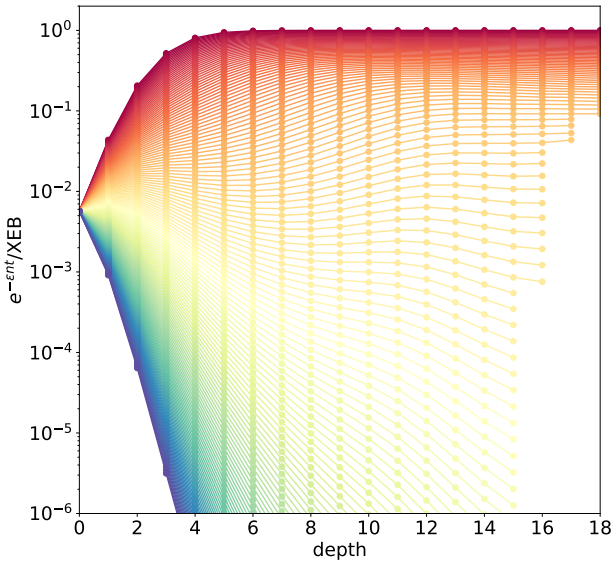


FIG. 13. Order parameter of the noise induced phase transition as a function of depth for different levels of noise for $0 \leq \epsilon n \leq 1.34$ (red to purple) on a $n = 18$ qubit chain.

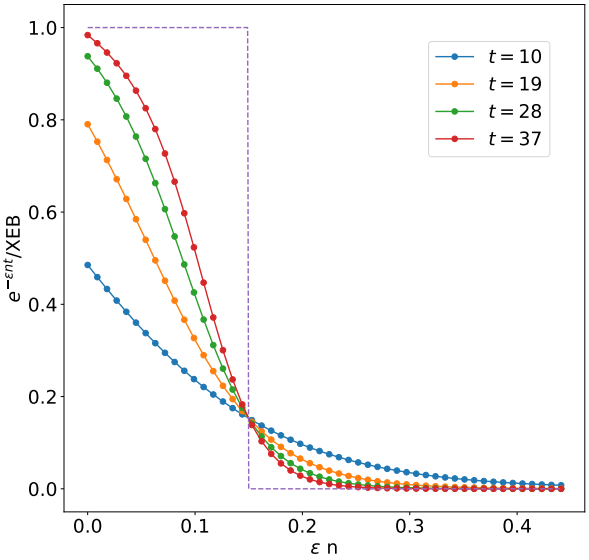


FIG. 14. Order parameter of the noise induced phase transition as a function of error per unit time for different depths on a $n = 18$ qubit chain. The link frequency is $1/T = 1/18$. The dashed line corresponds to the $d \to \infty$ limit.

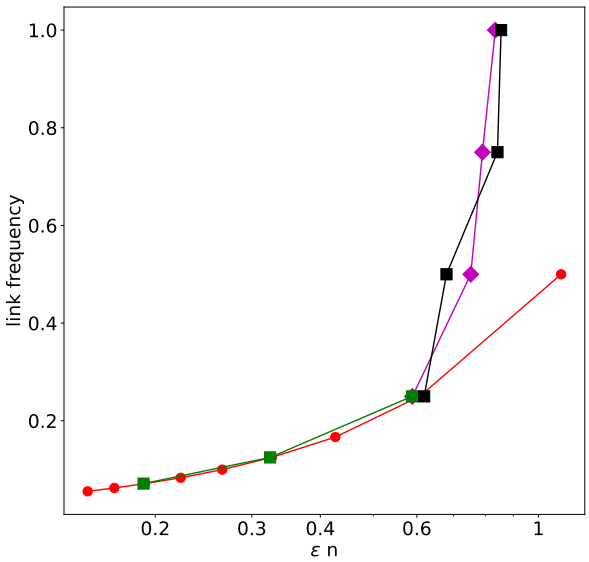


FIG. 15. Comparison of the transition point dependence on the weak link period T for the discrete gate set used in the experiment and uniformly random ensembles of single qubit gates. Black squares and magenta rhombus correspond to $n = 16$ (4×4) ABCD pattern for the experimental discrete gate set and continuous ensemble, respectively. Green squares and red circles correspond to $n = 16$ chain for the experimental and uniformly random gate set.

The role of the ensemble of single qubit gates does not appear to be significant for the NIPT point.

Appendix E: XEB phase diagram

1. XEB phase diagram in 1D

The population dynamics explained in Sec. D model the average dynamics for quantum circuits with a Markov chain. We focus on the case of uniformly random single qubit gate ensemble which results in a Markov chain with two states per site. In the absence of noise, this Markov chain has two steady configurations: the "thermal" or Porter-Thomas configuration; and the trivial "vacuum" configuration corresponding to the state normalization. After a short initial time, the state of a 1D circuit will be dominated by configurations with segments in the vacuum state and segments in the thermal configuration. We denote this configurations as $g_{\sigma_1,\ldots,\sigma_n}$ where $\sigma_k = \{0, 1\}$ denotes the vacuum and thermal state respectively.

This model is a generalization of the the weak-link model of Sec. D 3. Repeating the arguments of that section we obtain the following population dynamics update equation for multi-segment configurations

$$g_{\sigma_1,\ldots,\sigma_n}(d+1) = e^{-\epsilon \sum_i \sigma_i} 2^{-\sum_i (\sigma_i - \sigma_{i+1})^2} g_{\sigma_1,\ldots,\sigma_n}(d) \,. \quad \text{(E1)}$$

The factor $e^{-\epsilon}$ accounts for the fidelity decay with noise strength ϵ for a segment in the $\sigma = 1$ thermal state. The factor of $1/2$ between any two consecutive segments in different states comes from the application of an iSWAP, see Sec. D, which happens every two circuit cycles in 1D. This equation reproduces Eqs. (D10) to (D13) of the weak-link model with $n = 2$ and $T = 2$.

The initial state corresponds to an equal population of all configurations, $g_{\sigma_1,\ldots,\sigma_n} = 2^{-n}$, as obtained by generalizing the initial state given in Sec. D 2. The linear XEB at depth d, which we denote by XEB in this section, is given by

$$\text{XEB} + 1 = 2^n \sum_{\{\sigma_i\}} g_{\sigma_1,\ldots,\sigma_n}(d) \,. \quad \text{(E2)}$$

Equation (E1) can be solved for large n by a transfer matrix method. We will use the solution for a system of size n and depth d to express the solution of size $n + 1$ at the same depth d. Let's define

$$Z_n(\sigma, d) = 2^n \sum_{\{\sigma_i\}, i<n} g_{\sigma_1,\ldots,\sigma_{n-1},\sigma}(d) \,. \quad \text{(E3)}$$

From Eq. (E1) we obtain,

$$Z_{n+1}(\sigma, d) = e^{-\epsilon\sigma d} \sum_{\sigma_n = \{0,1\}} 2^{-(\sigma_n - \sigma)^2 d} Z_n(\sigma_n, d) \,. \quad \text{(E4)}$$

Equation (E4) must be solved with the boundary condition,

$$Z_1(\sigma, d) = e^{-\epsilon\sigma d} \,. \tag{E5}$$

The linear XEB is

$$\mathrm{XEB} + 1 = \sum_{\sigma} Z_n(\sigma, t) \,. \tag{E6}$$

We solve Eq. (E4) in the large n or continuous limit, approximating $Z_{n+1}(\sigma, d) \approx Z_n(d) + \partial_n Z_n(\sigma, d)$. Using the initial condition Eq. (E5) and substituting the result into Eq. (E6) we obtain,

$$\mathrm{XEB}(n, d) + 1 = 2e^{-\frac{\epsilon n d}{2}} \left(\cosh n\delta_d + \frac{2^{-d}}{\delta_d} \sinh n\delta_d \right), \tag{E7}$$

$$\delta_d \equiv \sqrt{\frac{\epsilon^2 d^2}{4} + 2^{-2d}} \,. \tag{E8}$$

Note that in the absence of noise $\epsilon \to 0$ we obtain

$$\mathrm{XEB}(n, d) + 1 = 2e^{n2^{-d}} \,. \tag{E9}$$

This result is consistent with Ref. [15] which shows that 1D random circuits anticoncentrate in logarithmic depth.

As explained in the main text, it is natural to introduce the scaling noise variable

$$f \equiv \epsilon n \,. \tag{E10}$$

To study the XEB phase diagram this variable is kept constant as $n \to \infty$. We also introduce the scaling depth variable

$$\alpha = \frac{d}{\log_2 n}. \tag{E11}$$

Including these substitutions the expression for XEB takes the form

$$\mathrm{XEB}(n; \alpha, f) + 1 = 2e^{-\frac{f\alpha \log_2 n}{2}} \left(\cosh \Delta + \frac{n^{1-\alpha}}{\Delta} \sinh \Delta \right) \,, \tag{E12}$$

where

$$\Delta \equiv \sqrt{\frac{f^2\alpha^2 \log_2^2 n}{4} + n^{2(1-\alpha)}} \,. \tag{E13}$$

We can now explain the different phases of XEB by writing the Eq. (E12) for fixed f and α in the thermodynamic limit $n \to \infty$. At low depth $\alpha < 1$, before anticoncentration, the second term under the square root in the Eq. (E13) for Δ dominates. We obtain

$$\mathrm{XEB}(n; \alpha, f) = n^{-\frac{f\alpha}{\ln 4}} \left(2e^{n^{1-\alpha}} \right) - 1 \,. \tag{E14}$$

This result differs from the noise-free limit Eq. (E9) by a noise-dependent algebraic prefactor whose exponent depends smoothly on noise.

After anticoncentration, $\alpha > 1$, the first term in Eq. (E13) dominates in the thermodynamic limit $n \to \infty$. We obtain

$$\mathrm{XEB}(n; \alpha, f) = n^{-\frac{f\alpha}{\ln 2}} + \frac{2n^{1-\alpha}}{f\alpha \log_2 n}, \tag{E15}$$

Clearly Eqs. (E14) and (E15) cannot be merged to each other by an analytic function, which indicates that $\alpha = 1$ is a phase transition line. Indeed for $\alpha = 1$ Eq. (E14) is algebraic whereas Eq. (E15) is logarithmic. This is in sharp contrast with the noise-free Eq. (E9) which does not have any singularity at $\alpha = 1$.

Equation (E15) describes another phase transition defined by the noise induced phase transition line

$$f_c(\alpha) = \frac{\alpha - 1}{\alpha} \ln 2 \,. \tag{E16}$$

This line separates the weak and strong noise regimes. Note that the depth required in 1D for entanglement to spread across all the qubits is $\alpha \simeq n/\log_2 n$, and then $f_c \simeq \ln 2$. In the weak noise regime $f < f_c$ we obtain the XEB

$$\mathrm{XEB}(n; \alpha, f) = n^{-\frac{f\alpha}{\ln 2}} = e^{-\epsilon n d} \,, \tag{E17}$$

which coincides with the circuit fidelity F^d. In the opposite regime $f > f_c$ we obtain the XEB

$$\mathrm{XEB}(n; \alpha, f) = \frac{2n^{1-\alpha}}{f\alpha \log_2 n} \,, \tag{E18}$$

which is much larger than the circuit fidelity.

2. XEB phase diagram in 2D and higher dimensions

The 1D update equation for multi-segment configurations Eq. (E1) can be generalized to higher dimensions as

$$g_{\sigma_1,\ldots,\sigma_n}(d+1) = e^{-\epsilon \sum_i \sigma_i} 4^{-\frac{1}{2\cdot\kappa} \sum_{\langle ij \rangle} (\sigma_i - \sigma_j)^2} g_{\sigma_1,\ldots,\sigma_n}(d). \tag{E19}$$

where $\langle i, j \rangle$ denotes the nearest neighbors on the D dimensional lattice and κ is the number of neighbors per qubit. The factor of $1/4$ in the second term comes from the application of an iSWAP, as in Eq. (E1) and Eqs. (D10) to (D13). The exponent of $1/4$ counts the number of segments in different states, devided by 2 because we count neighbors twice in the sum, and devided by κ because an ISWAP is applied between two segments every κ circuit cycles. The initial conditions and the expression of linear cross-entropy is the same as in $1D$ case.

The noise induced phase transition that we are interested in resides at $d \gg 1$. There we can expand the populations state around the two stationary configurations with all $\sigma_i = 0$ (vacuum) or all $\sigma_i = 1$ (thermal), using the so-called dilute flipped spin expansion. This gives the equation

$$\text{XEB} + 1 \simeq \sum_k \frac{1}{k!} e^{-\epsilon k d} \left(\frac{n}{4^d}\right)^k + \sum_k \frac{1}{k!} e^{-\epsilon (n-k) d} \left(\frac{n}{4^d}\right)^k , \quad \text{(E20)}$$

where the first and second term in the right-hand-side correspond to the expansion around $\sigma = 0$ and $\sigma = 1$ respectively, and the index k describes the total number of the flipped spins. The usual combinatorial factor $k!$ is needed to avoid the over-counting of same configurations. We can rewrite this equation as

$$\text{XEB} + 1 = \exp\left[e^{-\epsilon d}\left(\frac{n}{4^d}\right)\right] + e^{-\epsilon n d} \exp\left[e^{\epsilon d}\left(\frac{n}{4^d}\right)\right] , \quad \text{(E21)}$$

which generalizes Eq. (E8) to higher dimensions.

Similarly to the one dimensional case, we introduce the scaling variables

$$\alpha = \frac{d}{\log_4 n} , \quad f = \epsilon n . \quad \text{(E22)}$$

In order the study the XEB phase diagram, we consider the thermodynamic limit $n \to \infty$ at fixed α and f. We find

$$\text{XEB} + 1 = \exp\left[n^{1-\alpha} e^{-\frac{f\alpha \log_4 n}{n}}\right] + e^{-f\alpha \log_4 n} \exp\left[n^{1-\alpha} e^{\frac{f\alpha \log_4 n}{n}}\right] \quad \text{(E23)}$$

$$\simeq \exp\left(n^{1-\alpha}\right)\left(1 + e^{-f\alpha \log_4 n}\right) \quad \text{(E24)}$$

$$= \exp\left(n^{1-\alpha}\right)\left(1 + n^{-\frac{f\alpha}{\ln 4}}\right) . \quad \text{(E25)}$$

We note that the first factor

$$\exp\left(n^{1-\alpha}\right) = e^{n 2^{-2d}} \quad \text{(E26)}$$

describes the convergence to anticoncentration, in accordance to Eq. (E9) and Ref. [15]. The second factor is

$$1 + n^{-\frac{f\alpha}{\ln 4}} = 1 + e^{-\epsilon n d} . \quad \text{(E27)}$$

In contrast with the one dimensional case, Eq. (E25) does not have any singularity at finite f and $\alpha = 1$, and therefore does not exhibit a phase transition in the convergence to anticoncentration. We explain below how this phase transition appears due to a boundary effect.

The noise induced phase transition line at $\alpha > 1$ has a similar form to Eq. (E16),

$$f_c(\alpha) = \frac{\alpha - 1}{\alpha} \ln 4 , \quad \text{(E28)}$$

and terminates at $\alpha = 1$. The XEB value in the weak and strong noise regimes of this first order phase transition is similar to the one dimensional case

$$\text{XEB} = \begin{cases} n^{1-\alpha}, & f > f_c(\alpha) \\ n^{-\frac{f\alpha}{\ln 4}} = e^{-\epsilon n d}, & f < f_c(\alpha) \end{cases} \quad \text{(E29)}$$

In the weak noise regime $f < f_c(\alpha)$ the value of the linear XEB is the circuit fidelity, as expected. We can trace this contribution to the thermal or Porter-Thomas state in the second term of Eq. (E20). In the strong noise regime $f > f_c(\alpha)$ the value of the linear XEB is dominated by local correlations above the vacuum which we can trace to the first term in Eq. (E20). This is the situation studied in Refs. [14, 16, 17].

In the above analysis we neglected the existence of the boundary for a finite lattice. In the case of a regular lattice boundary qubits have a different number of nearest neighbours from the bulk qubits. As we will see, boundary qubits dominate the convergence to the thermal state and the behavior of XEB is qualitatively different.

We focus on the 2D case for simplicity. The presence of the boundary with r nearest neighbours adds the following boundary contribution to the XEB Eq. (E21)

$$\exp\left(e^{-\epsilon d} \frac{c_r \sqrt{n}}{4^{t \cdot r/4}}\right) , \quad \text{(E30)}$$

where $c_r \sqrt{n}$ is the lattice perimeter and the exponent $r/4$ in the $1/4$ factor accounts for the fact that in the boundary an iSWAP is applied every $r/4$ circuit cycles in 2D. In a regular 2D lattice each qubit in the boundary has $r = 3$ nearest neighbours, while in the Sycamore device lattice the number of neighbours in the boundary is $r = 2$. In the thermodynamic limit $n \to \infty$ at fixed α and f Eq. (E25) gets modified to

$$\text{XEB} + 1 \simeq e^{n^{1-\alpha} + c_2 n^{\frac{1-\alpha}{2}}} \left(1 + n^{-\frac{f\alpha}{\ln 4}}\right) . \quad \text{(E31)}$$

Remarkably, the boundary introduces a phase transition at $\alpha = 1$ even without noise, in contrast to the one dimensional case. This phase transition arises from the competition between the two terms in the exponent of the prefactor, $n^{1-\alpha}$ and $n^{(1-\alpha)/2}$. Note that Ref. [15], which studies the convergence to anticoncentration, considers circuits without boundaries that do not exhibit this phase transition. The noise induced phase transition gets displaced to

$$f_c(\alpha) = \frac{\alpha - 1}{\alpha} \ln 2 . \quad \text{(E32)}$$

Note that this is the same condition as the 1D case with boundary studied in the previous section.

As mentioned in the main text the noise induced phase transition is loosely analogous to Freederiks transitions in liquid crystals [18]. Here we clarify this analogy further. Freederiks transition is a result of a competition

between anchoring of the orientation of molecules in a nematic liquid crystal by its boundary and an electric field that tends to align the molecules in a perpendicular direction. The transition is controlled by a field magnitude that scales inversely proportionally to the size of the crystal. In the case of the noise induced phase transition this is analogous to the competition between a uniform configuration of the respective Ising model and some configuration that has a small number of domain walls. Furthermore the effect of noise is similar to a field, which induces the transition at the point that scales inversely proportionally to the system size.

Appendix F: Noisy phase transition and spoofing

The existence of the noise induced phase transition in the noisy random circuit dynamics discussed in the main text has important implications for the effectiveness of the so called spoofing algorithms [14, 16, 17]. Indeed, building up this theory, we give below a lower bound for the error per cycle below which spoofing algorithms can not match the experimental XEB. This boundary can also be interpreted as a first order phase transition.

Spoofing algorithms aim to generate bitstrings from a distribution that maximizes the value of XEB without resorting to the exponential cost of full simulation of the quantum evolution, i.e. ideally with polynomial cost. These algorithms can be broadly split into two stages [14]: (i) approximation of the wave function, (ii) sampling of bitstrings to maximize XEB. We analyze step (i) first.

1. Spoofing in the weak-link model

Consider the weak-link model of Eq. (2) of the main text. The spoofing in this case corresponds to the elimination of entangling gates across the weak link. The omission of an iSWAP gate introduces a new two qubit operation into the population dynamics, corresponds to replacing $\hat{\Omega}^{(i,j)} \to \hat{\Omega}^{(i,j)}_{\text{omitted}}$ in Eq. (D4),

$$\hat{\Omega}^{(i,j)}_{\text{omitted}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} \end{pmatrix}. \tag{F1}$$

The system of equations for the dynamics of the weak-link model Sec. D 3, Eq. (D13) is modified as follows,

$$g_{00}(d+T) = g_{00}(d), \tag{F2}$$

$$g_{01}(d+T) = \frac{1}{4} g_{01}(d), \tag{F3}$$

$$g_{10}(d+T) = \frac{1}{4} g_{10}(d), \tag{F4}$$

$$g_{11}(d+T) \simeq \frac{1}{4} g_{11}(d). \tag{F5}$$

Notice that g_{11} is no longer constant even at $F = 1$. The resulting linear XEB after mT cycles reads

$$\langle D p_{\text{sim}}(s) - 1 \rangle_{\text{spoof}} = 3 \frac{1}{4^m}. \tag{F6}$$

We compare this to XEB of the noisy circuit with system fidelity $\approx F^{mT}$. Note that if the fidelity per cycle is smaller than $F^T_{cs} = 1/4$, the classical algorithm that omits the iSWAP gate in the weak link obtains a larger XEB value. This is different from the transition point of the corresponding noise induced phase transition, $F^T_c = 1/16$ from Eq. (2) of the main text. This means that XEB is a good estimator of fidelity already for $F > F^T_c$, but nonetheless a spoofing algorithm could produce an XEB higher than experiment while $F \leq F^T_{cs}$.

2. Spoofing for general models

The result of Eq. (F6) can be extended naturally to general spoofing algorithms and models using the formalism of population dynamics, see Sec. D. Spoofing algorithms can exploit two different contributions to cross entropy. The first is given by the small potential overlap that spoofing can still maintain with the ideal Porter-Thomas state, including some amount of global correlations. The ideal Porter-Thomas state is given by g_{11} in the weak-link model above, and this contribution is modelled as removing two-qubit gates, as in Eqs. (F1) and (F5). In the general case, we can break the full circuit into subsystems, each with a manageable number of qubits, typically $\sim n/2$ [6, 14, 19]. This is done by removing two-qubit gates along the cut that separates both subsystems. Each iSWAP gate removed lowers this contribution to XEB by a factor of 1/4 [6, 19]. This contribution then scales as $1/4^{\nu d}$, where ν is the number of gates along the cut, and d is the depth. Given the significant number of gates along any suitable cut [6], this contribution is subdominant compared to the second contribution which we study next.

The dominant contribution from spoofing algorithms to XEB is that they can potentially capture finite-depth local correlations outside the thermal or Porter-Thomas state [14, 16, 17]. In the weak-link model above, these finite depth local correlations are modelled by the vacuum-"thermal" and "thermal"-vacuum states, g_{01} and g_{10}. Their exponential decay for increasing depth is given in this model by Eqs. (F3) and (F4).

In the general case, the thermal or Porter-Thomas state is the stationary eigenstate of a Markov chain modeling the dynamics when averaging over circuits, as explained in the population dynamics Sec. D. The magnitude of finite-depth local correlations outside of the Porter-Thomas state can be bounded by the difference between the noiseless XEB value at finite depth and the asymptotic value. This gives the equation

$$\langle D p_{\text{sim}}(s) - 1 \rangle_{\text{spoof}} \lesssim \text{XEB} - C(n)\,, \tag{F7}$$

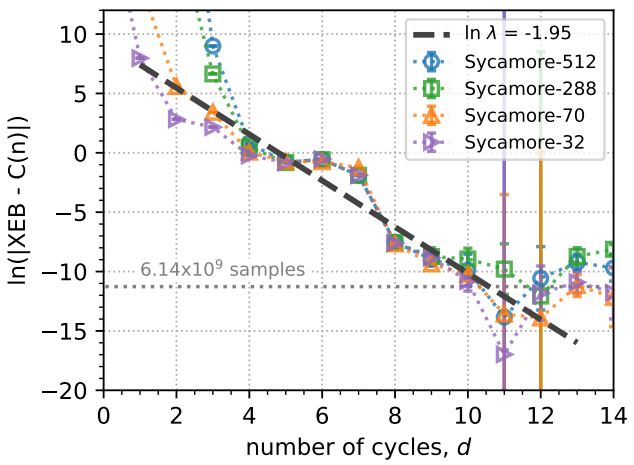


FIG. 16. Logarithm of $\mathrm{XEB} - C(n)$ as a function of the number of cycles d for random noise-free Clifford circuits. Different colors correspond to different number of qubits. For all simulations, the Sycamore layout and the pattern ABCD-CDAB is used. Here $C(n)$ is the asymptotic value of the XEB for n qubits. The horizontal dotted gray line corresponds to scale of the numerical error from the statistics resolution limit of $1/\sqrt{M}$, where $M = 6.14 \times 10^9$ is the number of samples used.

where

$$C(n) = \frac{1 - 2^{-n}}{1 + 2^{-n}} \simeq 1 \tag{F8}$$

is the asymptotic value of XEB for given n. The XEB value will converge exponentially with depth as

$$\mathrm{XEB} - C(n) \propto \lambda^d \,. \tag{F9}$$

We extract the decay rate λ from numerical simulations, see Fig. 16. We obtain a decay rate $\lambda \approx e^{-1.95}$. This is a much faster decay rate than the decay of fidelity observed in the experiments reported here. Note that we omit the non-Clifford gates for the numerics used in Fig. 16 so we can scale up to 512 qubits. This can be done because it does not affect the average value of XEB for the ensemble of circuits, as shown in Sec. H 5. The discrete gate set used here, together with the iSWAP gate [4, 13, 19, 20], gives a faster convergence than other gate sets. In particular, we obtain a faster convergence than the estimation of Sec. E 2 which uses a uniformly random single qubit gate ensemble [11, 12].

As we will see below, the post-processing stage (ii) of a spoofing algorithm can only affect a multiplicative factor, and not the exponential decay rate.

3. Linear XEB amplification with post-processing

A specific spoofing algorithm discussed in Ref. [14] approximates in step (i) the wave function by taking a product of two subsystem wave functions, each of size roughly half of the full system. This corresponds to omitting from the circuit all gates entangling the two subsystems. In the post-processing step (ii) the output bitstrings are chosen to maximize the approximate probabilities, instead of sampling the resulting approximate wave function. In Ref. [14] an average XEB corresponding to this procedure was estimated numerically.

The post-processing stage (ii) in Ref. [14] is done sorting the probabilities within each subsystem, which incurs an exponential cost $\sim 2^{n/2}$. This is doable for relatively small subsystems. The number of distinct bitstrings to be produced has to match the number in the experiment $k \sim 10^6 \ll D$, see Sec. I 2.

In what follows we show that the linear XEB is upper bounded by

$$\langle D p_{\mathrm{sim}}(s) - 1 \rangle_{\mathrm{spoof}} \lesssim \ln(D_L/k_L) \ln(D_R/k_R) \times \lambda^d, \tag{F10}$$

where D_L and D_R are the Hilbert space dimensions for the left and right subsystems respectively, and k_L and k_R are number of samples from left and right subsystems. Note that $D = D_L D_R$ and $k = k_L k_R$. It is important to emphasize that the enhancement of XEB is at most logarithmic in the size of the Hilbert space of each subsystem (linear in the respective number of qubits). Contemplating equation (F10) it may be tempting to split the Hilbert space into $r > 2$ subsystems and expect an enhanced factor $\propto (\ln D^{1/r})^r$. However, while using more subsystems could in principle increase the post-processing multiplicative prefactor, it would result in a worse approximation because it ignores correlations between the subsystems.

For the parameters of the experiment in Fig. 4 of the main text, $d = 24, n = 70$ and $\ln \lambda \simeq -1.95$, the upper bound on spoofing XEB (F10) is well below the value in the experiment. This indicates that spoofing cannot be successful.

We now describe the derivation of the optimal the pre-exponential multiplicative factor in Eq. (F10). Consider M random numbers $w_i, i = \{1, ..., M\}$ sampled from a joint probability density $\mathcal{P}(w_1, w_2, ..., w_M)$. Consider the sorted list of $w_i : w_1 > w_2 > ....$ The probability density for the kth term in the list, or k-th order statistic, is

$$\mathcal{P}_k(w) = \frac{1}{(k-1)!(M-k)!} \sum_S \prod_{r=1}^{k-1} \int_w^\infty dw_r \prod_{q=k+1}^{M} \int_0^w dw_q \mathcal{P}(S(w_1, w_2, ..., w_M)), \tag{F11}$$

where the sum is over all permutations of w_i. Identifying w_j with a bitstring probability and M with the dimension of the respective Hilbert space, the XEB is given by the following average,

$$\langle w \rangle = \frac{1}{k} \sum_{k'=1}^{k} \langle w \rangle_{k'}, \tag{F12}$$

$$\langle w \rangle_{k'} = \int_0^\infty dw w \mathcal{P}_{k'}(w). \tag{F13}$$

At the depths considered experimentally, each subsystem reaches the Porter-Thomas or exponential distribution. Then the corresponding k-th order statistic has probability density

$$\mathcal{P}_k(w) = \frac{D!}{(k-1)!(D-k)!} D e^{-kDw} \left(1 - e^{-Dw}\right)^{D-k}. \tag{F14}$$

Substituting this into the expression for $\langle w \rangle$, Eq. (F13), we find for $k \gg 1$,

$$D\langle w \rangle = \ln(D/k). \tag{F15}$$

This expression is the origin of the logarithmic factors in Eq. (F10). Note that is natural to expect this, because the probability to find a state on the tail of probability distribution decays exponentially.

This is the optimal enhancement from the stage (ii) of the spoofing algorithm if stage (i) is able to reproduce the wave function of each subsystem. Therefore we obtain

$$\langle D p_{\rm sim}(s) - 1 \rangle_{\rm spoof} \lesssim \lambda^d D \frac{1}{k} \sum_{k',k''} \langle w \rangle_{k'}^L \langle w \rangle_{k''}^R, \tag{F16}$$

where the sum over k', k'' includes k terms in total and is defined to maximize the right hand side of Eq. (F16). Using Eq. (F16) we find that the maximum corresponds to a linear configuration such that $\ln D_L/k_L = \ln D_R/k_R$.

4. Logarithmic XEB amplification with post-processing

It is worthwhile to consider the spoofing of the logarithmic XEB, see Eq. (A17) for a formal definition. Logarithmic XEB is less sensitive to rare spikes of the wave function amplitude in the bistrsing basis.

The first step of the XEB amplification in Ref. [14], and in the previous section, is to devide the system in two subsections. The ideal wave function is a superposition of products of left $|\psi_{L,i}\rangle$ and right $|\psi_{R,i}\rangle$ wave functions [19],

$$|\psi\rangle \simeq \frac{1}{\sqrt{\mathcal{N}}} \sum_i |\psi_{L,i}\rangle |\psi_{R,i}\rangle . \tag{F17}$$

Using the analysis similar to the previous subsection we obtain a bound on logarithmic XEB after stage (ii) of the spoofing algorithm

$$\log \text{XEB}_{\rm spoof} \simeq \ln \left[1 + \frac{1}{\mathcal{N}} \ln(D_L/k_L) \ln(D_R/k_R) \right]. \tag{F18}$$

Note that for large $\mathcal{N} \gg 1$ logarithmic XEB of the bitstrings produced by spoofing gives the same result as linear XEB.

Appendix G: Simulation of random circuit sampling using tensor network contraction

Tensor network contraction has been used extensively in the simulation of RCS over the last few years [21–28]. Given a quantum circuit, it is straightforward to generate a tensor network whose contraction yields one or many of its output amplitudes. In this tensor network, each one-qubit gate is expressed by a rank-2 tensor, each two-qubit gate is expressed by a rank-4 tensor, and the input state $|0\rangle^{\otimes n}$ is expressed by the tensor product of n rank-1 tensors.

The time and memory complexities of the contraction of such a tensor network depend strongly on the order in which tensors are contracted. Its time complexity has a lower bound related to the treewidth of the line graph of the tensor network [21]. Ref. [29] introduced a method to alleviate the memory requirements for the contraction of a tensor network at the expense of a larger time complexity. This method involves "slicing" (that is, projecting) certain carefully chosen indices in the tensor network to the different values in their support. Each slice yields a tensor network that requires less memory to be contracted, although one has to contract a number of tensor networks that scales exponentially in the number of indices sliced. Refs. [23, 24] made substantial improvements in the optimization of contraction orderings and choice of slices.

Slices are useful to lower the memory requirement of a large tensor network contraction. In the context of simulating RCS, slices can also help reduce the computation time while reducing the fidelity of the output state.

This was introduced in Ref. [19], where it was shown that summing over a fraction f of slice or projection instances results in an output state of fidelity f with respect to the $f = 1$ ideal state. This, naturally, results in a decrease of the computation time by a factor f. This method was validated in more generic settings in Refs. [30] and [24] and we apply it in our time estimates for a target fidelity equal to the estimated experimental fidelity.

Works described up to this point mostly focused on the contraction of an independent tensor network per output bitstring, which results in a simulation runtime that scales linearly in the number of bitstrings sampled. Specifically, an algorithm to sample from the output distribution of a quantum circuit is as follows: 1) sample bitstrings uniformly at random; 2) calculate the ideal probabilities for these bitstrings; 3) perform rejection sampling to select a subset of these bitstrings as the output. The frugal rejection sampling proposed in Ref. [19] requires computing only about 10 probabilities per output bitstring sampled. We can calculate the probabilities of $\gtrsim$10 very similar bitstrings with just one contraction, and as we will only select one using rejection sampling and the probabilities are still uncorrelated, the result is the same as the algorithm above [30].

Ref. [27] introduced a method to compute amplitudes of a large number of uncorrelated bitstrings with a much lower overhead than linear. This implies the sparsification of the output of the tensor network as output tensors are being contracted: only those tensor entries that will lead to the computation of an amplitude of a bitstring in a pre-specified set are kept. In addition, Ref. [27] managed to make use of a special property of the fSim gate [6] in order to propagate the slicing of certain indices to other "related" indices at no extra cost. These two advancements allowed Ref. [27] to simulate sampling 1 million uncorrelated bitstrings from the largest circuit of Ref. [6] in 15 hours using 512 NVIDIA Tesla V100 SXM3 GPUs with 32GB of RAM each.

Similarly, Ref. [26] introduced dynamic programming techniques to reuse certain computations across the different instances of the slices taken over a tensor network. This allowed them to simulate the largest circuit of Ref. [6] in 14.5 days using 32 NVIDIA Tesla V100 GPUs with 16GB of RAM each.

In the present work we incorporate all of these advancements into a highly efficient simulated annealing optimizer in order to further reduce the time estimates for the simulation of RCS experiments. This goal is achieved by co-optimizing both the tensor network contraction ordering and the slices to use to reduce the memory footprint of the contraction.

In our protocol, the tensor network contraction is represented as an ordered list $\mathcal{I}$ of indices in the tensor network, which are contracted from the first to the last. If an index is contracted between two tensors, all indices between them are then contracted at the same time (regardless of their position in $\mathcal{I}$). This ensures that, for any $\mathcal{I}$, there is one and only one tensor network contraction (however, multiple $\mathcal{I}$ may correspond to the same contraction). Similarly, indices that are sliced are represented as an ordered list $\mathcal{S}$ of indices of the tensor network and a number α. That is, an index i in the tensor network is sliced only if it appears in $\mathcal{S}$ at a position $\alpha_i \leq \alpha$.

At the beginning of the protocol, a random ordering of the indices $\mathcal{I}$ is chosen, as well as a random ordering of indices $\mathcal{S}$. At any moment of the simulation, it is always guaranteed that α is large enough to allow the tensor network contraction induced by $\mathcal{I}$ to fit in memory.

The co-optimization of $\mathcal{I}$ and $\mathcal{S}$ is done in steps, with $\mathcal{I}$ being optimized at every step (while $\mathcal{S}$ is kept constant), and $\mathcal{S}$ at every 10 steps (while $\mathcal{I}$ is kept constant). A move for $\mathcal{I}$ consists in swapping two randomly chosen indices in $\mathcal{I}$, which will induce a new contraction ordering. Similarly, a move for $\mathcal{S}$ consists in swapping two indices in $\mathcal{S}$, and α is chosen as the smallest position such that the contraction ordering induced by $\mathcal{I}$ fits in memory. The indices swapped are chosen at random, one of them from the set of indices with $\alpha_i \leq \alpha$ and the second one from the set of indices with $\alpha_i > \alpha$. Since increasing α can only reduce the memory footprint, α can be efficiently found by bisection.

Every time $\mathcal{I}$ or $\mathcal{S}$ are changed, the new FLOP count (including the overhead induced by the slicing) is computed. The FLOP count is computed by creating a "slicing tree" where each node correspond to a bifurcation created by slicing a given index. Therefore, the slicing tree will have α levels and 2^α leaves, with each leaf corresponding to a specific projection of indices. The order of the slicing tree is determined by the order in which they appear in the contraction ordering induced by $\mathcal{I}$, with the root being the first index sliced. To reduce the computational cost, every time there is a bifurcation, the state of the contraction (including any intermediate tensors) is cached (checkpoint). When a new projection is computed, it is enough to unwind the contraction up to the last checkpoint, avoiding the calculation of intermediate tensors that are guaranteed to be the same. Once the new FLOP count is computed, the move is accepted by using the Metropolis-Hastings algorithm, with an inverse temperature linearly changing from 10^{-5} to 10^5.

Let us consider now the set $\mathcal{M}$ of uncorrelated bitstrings to sample. Each tensor can have "internal" indices (i.e., indices that eventually are contracted) and "sparse" indices. Such sparse indices will not have all the possible combination of values, but only those combinations compatible with $\mathcal{M}$ (i.e., only valid projected bitstrings). Therefore, if a tensor has k internal indices and m sparse indices, its memory footprint can not be larger than (and, in the worst, case equal to) $2^k \min\{2^m, |\mathcal{M}|\}$. In our protocol, we always assume the worst case scenario.

In a similar way, we can compute the cost of contracting two tensors with sparse indices. More precisely, the contraction cost will be equal to the contraction cost without any sparse indices, multiplied by the total number of compatible projections that can be created com-

bining the sparse indices of the two tensors. Note that the calculation of the exact contraction cost involves the knowledge of $\mathcal{M}$ and the projected bitstrings in both contracted tensors. However, keeping track of all the projected bitstrings is computational demanding. To avoid this bottleneck, we approximate the number of combined projected bitstrings as $\min\{2^{m'}, |\mathcal{M}|\}$, where m' is the number of sparse indices in the tensor resulting from the contraction of the two initial tensors. While this approximation does not take into account the structure of $\mathcal{M}$, it gives an asymptotically tight upper bound of the actual cost.

All these techniques allow us to reduce the number of FLOPs required for the noisy simulated sampling of 1 million bitstrings from the hardest circuit of Ref. [6] by about an order of magnitude compared to the requirements of Refs. [27] and [26] when using a similar cluster with similar GPUs. When running on a Google Cloud CPU with 12 TB of memory, we estimate FLOP counts about two orders of magnitude lower than in Refs. [27] and [26]. We estimate a runtime of 2 days using a single CPU with a 20% FLOP efficiency, similar to the efficiency found in Refs. [23, 24, 27]. Table I of the main text shows the runtime estimates for the simulation of the experiments of Refs. [6, 31, 32] and the present work when run on the Frontier supercomputer.

Appendix H: Bounds to approximate tensor representations

The most promising numerical methods for more efficient approximations to random circuit sampling are based on approximate tensor representations [23, 33, 34]. Let's write the state on n qubits as

$$|\psi\rangle = \sum_{j_1,j_2,\ldots,j_n} \psi_{j_1,j_2,\ldots,j_n} |j_1, j_2, \ldots, j_n\rangle \ . \tag{H1}$$

We study the simplest but illustrative case where the state is approximated with two tensors $M^{(1)}$ and $M^{(2)}$ as [34]

$$\tilde{\psi}_{j_1,j_2,\ldots,j_n} = \sum_{\alpha=1}^{\chi} M^{(1)}_{j_1,j_2,\ldots,j_l,\alpha} M^{(2)}_{\alpha,j_{l+1},j_{r+2},\ldots,j_n} \ . \tag{H2}$$

The index α is called a virtual index and χ is called the bond dimension. The state is broken into qubits on the left $[1\ ..\ l]$ and the right $[r+1\ ..\ n]$, with the virtual index encoding the entanglement between these spaces. The size of the two sub-Hilbert spaces are respectively $D_1 = 2^l$ and $D_2 = 2^{n-l}$.

Given a quantum state $|\psi\rangle$ and bond dimension χ, the best approximation of the form (H2) can be found keeping the largest χ singular values (Schmidt coefficients) of a matrix with entries corresponding to the amplitudes of $|\psi\rangle$, and row and column dimensions corresponding to the left and right spaces. This gives the approximation

$$|\tilde{\psi}\rangle = \frac{1}{\sqrt{\sum_{\alpha=1}^{\chi} S_\alpha^2}} \sum_{\alpha=1}^{\chi} S_\alpha |l_\alpha\rangle |r_\alpha\rangle \,, \tag{H3}$$

based on the Schmidt decomposition, where the Schmidt coefficients S are ordered from large to small. This representation is exact if the bond dimension is larger than the Schmidt rank.

For approximate matrix-product state (MPS) simulations [33, 34], the initial state is a product state with Schmidt rank equal zero. Gates involving only qubits on the left (or right) merely modify the tensor $M^{(1)}$ (or $M^{(2)}$) without affecting the Schmidt rank. However, multi-qubit gates applied on qubits belonging to both left and right partition, generally increase the Schmidt rank. The truncation of the quantum state to a fixed bond dimension $\chi \ll 2^{\min(l,n-l)}$ eventually reduces the fidelity to $F = |\langle\psi|\tilde{\psi}\rangle|^2 = \sum_{\alpha=1}^{\chi} S_\alpha^2$.

We quantify the fidelity for random Haar states in Sec. H 1 and arbitrary states in Sec. H 2. We also show that linear XEB remains a good estimator of fidelity in this case, see Sec. H 4 [35]. We provide both numerical and analytical bounds on the bond dimension χ required to achieve a given fidelity for the simulation of RCS using MPS methods. We give a precise method to compute this lower bound, see Sec. H 5. We also pinpoint the number of cycles of a sharp transition to the "typical" (quasi-maximum) entanglement in Sec. H 6 (see also Ref. [36]).

In order to be of any practical use, χ must be much smaller than the Hilbert space dimension of the halves, with $\chi \lesssim 10^3$ for most realistic implementations. Fig. 22 shows the required χ to reach a fidelity of $F \approx 10^{-4}$, as a function of the number of qubits and cycles. The reported memory footprint is the required memory to store two complex arrays of sizes $2^{n/2} \times \chi$. For 70 qubits and 24 cycles, the bond dimension χ required is of the order of 10^7 (with 35 qubits in each half), which is well beyond the capacity of Frontier.

1. Fidelity for Haar random states

The output state of a sufficiently deep quantum random circuit can often be approximated as a Haar random state. Therefore, we can explain the fidelity obtained with representation (H2) using the distribution of singular values of a complex matrix with Gaussian entries [3] and dimension $D_1 \times D_2$, where D_1 (D_2) is the dimension of the left (right) Hilbert space. We extend the study of Ref. [34] to the case $D_1 \neq D_2$, and we assume without loss of generality $D_1 \leq D_2$.

Let S denote singular values and $s = \sqrt{D_1} S$ denote normalized singular values. The distribution of the normalized singular values follows the Marčenko–Pastur dis-

tribution [37]

$$p_\lambda(s) = \frac{1}{\pi}\frac{\sqrt{(\lambda_+^2 - s^2)(s^2 - \lambda_-^2)}}{\lambda s}, \tag{H4}$$

with $D_1 \leq D_2$, $\lambda = D_1/D_2$, $\lambda_\pm = (1 \pm \sqrt{\lambda})$ and $s \in [\lambda_-,\ \lambda_+]$. For $\lambda = 1$, $p_\lambda(s)$ reduces to:

$$p(s) = \frac{1}{\pi}\sqrt{4 - s^2}. \tag{H5}$$

For a given bond dimension χ the fraction of singular values is $r = \chi/D_1$. Let us define the cumulative distribution of singular values

$$C_\lambda(s') = \int_{s'}^{\lambda_+} ds\, p_\lambda(s) \tag{H6}$$

Note that $r = C_\lambda(s')$ is the fraction of singular values up to a given normalized singular value s', starting from the largest singular values. Therefore, the normalized singular value corresponding to a given fraction r is:

$$s_\lambda(r) = C_\lambda^{-1}(r). \tag{H7}$$

For $\lambda = 1$ one obtains [34]

$$s_{\lambda=1}(r) = 2\cos\left(\frac{1}{2}\mathcal{A}^{-1}(\pi r)\right)\ , \tag{H8}$$

where $\mathcal{A}(\theta) = \theta - \sin\theta$.

The fidelity for a given fraction r of singular values is given by

$$\mathcal{F}_\lambda(r) = \int_{s(r)}^{\lambda_+} ds\, s^2 p_\lambda(s). \tag{H9}$$

Recalling that

$$\frac{ds_\lambda(r)}{dr} = \frac{dC_\lambda^{-1}(r)}{dr} = \frac{1}{\frac{dC_\lambda}{ds}\big|_{s=s_\lambda(r)}} = -\frac{1}{p_\lambda(s_\lambda(r))},$$

one also gets an alternative expression

$$\begin{aligned}\mathcal{F}_\lambda(r') &= -\int_{r'}^{0} dr \frac{1}{p_\lambda(s_\lambda(r))} s_\lambda^2(r)\, p_\lambda(s_\lambda(r)) \\ &= \int_0^{r'} dr\, s_\lambda^2(r)\ . \end{aligned} \tag{H10}$$

For $\chi \ll D_1$ we have

$$\mathcal{F}_\lambda(\chi/D_1) \leq \left.\frac{d\mathcal{F}_\lambda(r)}{dr}\right|_{r=0} \frac{\chi}{D_1} = \lambda_+^2\,\frac{\chi}{D_1}, \tag{H11}$$

where we used the fact that $\mathcal{F}_\lambda$ is a monotonically increasing and strictly concave function (that is, $d^2\mathcal{F}_\lambda/dr^2 = -2s_\lambda(r)/p_\lambda(s_\lambda(r)) \leq 0$). For $\lambda = 1$ [34], the bound reduces to $\mathcal{F}_{\lambda=1} \leq \frac{4\chi}{\sqrt{D}}$. This gives an upper bound for the fidelity of a single projection into the ansatz of Eq. (H2) once the ideal state is sufficiently entangled.

2. Fidelity bound for arbitrary states

The fidelity of a Schmidt-decomposed state as in Eq. (H3) is

$$F = \sum_{\alpha=1}^{\chi} S_\alpha^2. \tag{H12}$$

Using the Jensen's inequality, it follows that

$$\chi^2\left(\frac{1}{\chi}\sum_{\alpha=1}^{\chi} S_\alpha^2\right)^2 \leq \chi \sum_{\alpha=1}^{\chi} S_\alpha^4 \tag{H13}$$

and therefore

$$F \leq \sqrt{\chi \sum_{\alpha=1}^{\chi} S_\alpha^4} \leq \sqrt{\chi\ \mathrm{tr}\,\rho_L^2}, \tag{H14}$$

with $\mathrm{tr}\,\rho_L^2 = \sum_{\alpha=1}^{D_1} S_\alpha^4$ being the reduced purity after tracing out the qubits on the right. Using exact numerics for small systems (see Fig. 17), one can find a tighter bound

$$F \lesssim \mathcal{F}_\lambda\left(\chi\,\mathrm{tr}\,\rho_L^2\right) \leq \lambda_+^2\,\chi\,\mathrm{tr}\,\rho_L^2, \tag{H15}$$

with $\mathcal{F}_\lambda$ being the fidelity for Haar random states, see Eq. (H10), and the average is at fixed depth. Equations (H14) and (H15) gives an upper bound for the fidelity of a single projection for an arbitrary quantum state in the form (H2).

We can compare the numerical bound Eq. (H15) with the fidelity for Haar random states Eq. (H11) for $D_1 = \sqrt{D}$. The average purity for a Haar random state when $D_1 = \sqrt{D}$ is:

$$\begin{aligned}\langle\langle \mathrm{tr}\,\rho_L^2\rangle\rangle &= \sum_{\alpha=1}^{D_1}\langle\langle S_\alpha^4\rangle\rangle = \frac{1}{D_1^2}\sum_{\alpha=1}^{D_1}\langle\langle s_\alpha^4\rangle\rangle \\ &= \frac{1}{D_1}\int_0^2 s^4\frac{1}{\pi}\sqrt{4-s^2}ds = \frac{2}{\sqrt{D}}\ . \end{aligned} \tag{H16}$$

Therefore, for small $\chi/\sqrt{D}$, the numerical bound Eq. (H15) is only twice larger than the correct fidelity in this case.

3. Open and close simulations using approximate tensor representations

As described in [34], approximated tensor representations can be used to sample bitstrings with a given target fidelity F. More precisely, authors of [34] present two different protocols: "open" and "close" simulations.

For open simulations at fixed bond dimension χ, the circuit C is split in k sub-circuits such that $C = C_k\ \cdots\ C_1 C_0$. Starting from the initial state $|\psi_0\rangle = |0\rangle$,

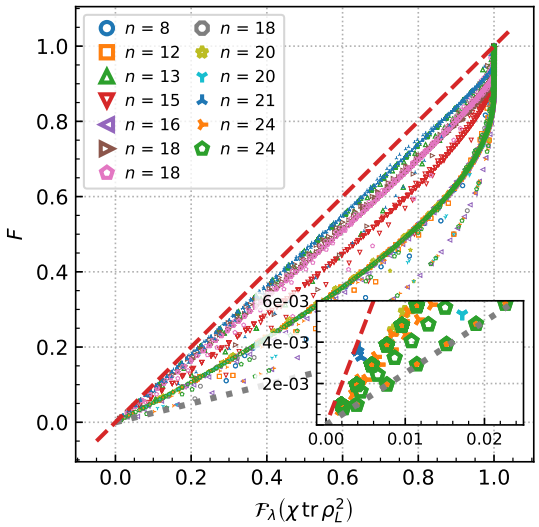


FIG. 17. Numerical fidelity bound. The plot compares the exact fidelity F for different Sycamore layouts of different sizes, depths, and bond dimensions χ to the numerical upper-bound $\mathcal{F}_\lambda\left(\chi \operatorname{tr}\rho_L^2\right)$, with $\mathcal{F}_\lambda$ and $\operatorname{tr}\rho_L^2$ being respectively the fidelity in the Haar limit and the reduced purity. The dotted-gray line corresponds to the bound $\mathcal{F}_\lambda\left(\chi \operatorname{tr}\rho_L^2\right) \leq \lambda_+^2 \chi \operatorname{tr}\rho_L^2 \leq 4\chi \operatorname{tr}\rho_L^2$. For all the instances, the pattern ABCDCDAB is used, and the qubits are partitioned in two equal halves using a diagonal cut.

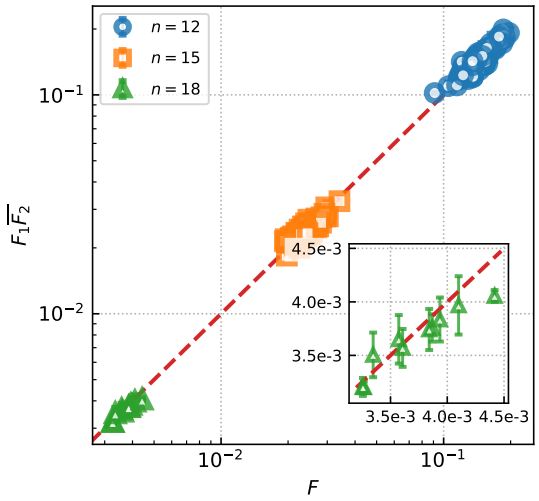


FIG. 18. Fidelity for close simulations $F_1\overline{F_2}$ (see App. H 3), compared to the exact fidelity F. The bond dimension is fixed to $\chi = 8$. Each point correspond to a different circuit, with the average being performed over bitstrings. Circuits are split in three parts of $|C_1| = 8$, $|C_M| = 4$ and $|C_2| = 8$ cycles respectively. For all the instances, the pattern ABCDCDAB is used, and the qubits are partitioned in two equal halves using a diagonal cut.

a new approximate state $|\tilde{\psi}_1\rangle$ is obtained by truncating the state $|\psi_1\rangle = C_1 |\psi_0\rangle$ using its χ largest singular values $\{S_{\alpha,1}\}$ only. The approximate state $|\tilde{\psi}_1\rangle$ is then evolved to get $|\psi_2\rangle = C_2 |\tilde{\psi}_1\rangle$, which is again truncated to its largest χ singular values to obtain the approximate state $|\tilde{\psi}_2\rangle$. Calling $f_k = |\langle\psi_k|\tilde{\psi}_k\rangle|^2 = \sum_{\alpha=1}^{\chi} S_{\alpha,k}^2$ the partial fidelity of the approximate state $|\tilde{\psi}_k\rangle$, Ref. [34] shows that, for random circuits, the final fidelity can be expressed as the product of all the partial fidelity, that is $F = |\langle\tilde{\psi}_k|C|\psi_0\rangle|^2 = f_1 f_2 \cdots f_k$.

For a large number of qubits, computing $|\psi_i\rangle = C_i |\psi_{i-1}\rangle$ and finding its singular values to get the approximate state $|\tilde{\psi}_i\rangle$ is numerically intractable. To overcome this limitation, authors of [34] introduce a variational approach to compute $|\tilde{\psi}_i\rangle$ without the need of the intermediate state $C_i |\tilde{\psi}_{i-1}\rangle$ and without explicitly computing its singular values. More precisely, for any C_i, a new random approximate tensor $|\tilde{\phi}_i\rangle$ of bond dimension χ is used to compute the objective:

$$\tilde{f}_i(|\tilde{\phi}_i\rangle) = |\langle\tilde{\psi}_{i-1}|C|\tilde{\phi}_i\rangle|^2. \tag{H17}$$

Recalling that both $|\tilde{\psi}_{i-1}\rangle$ and $|\tilde{\phi}_i\rangle$ are MPS of bond dimension χ, computing $\tilde{f}_i$ for sufficiently shallow C_i and small χ is doable even for a large number of qubits [34]. Using the update strategy proposed in [34], one can find $|\tilde{\psi}_i\rangle$ as

$$|\tilde{\psi}_i\rangle = \underset{|\tilde{\phi}_i\rangle}{\operatorname{argmax}} \tilde{f}_i(|\tilde{\phi}_i\rangle). \tag{H18}$$

Because the quantum system becomes more and more entangled by applying the sub-circuits C_i, one expects that $f_{i+1} \leq f_i$, reaching the saturation value of $\mathcal{F}_\lambda(\chi/D_1)$ when the quantum state reaches the random Haar limit.

Close simulations at fixed bond dimension χ are useful to sample bitstrings with an improved target fidelity than the corresponding open simulations with the same bond dimension. To start, the circuit is split in three parts $C = C_2 C_M C_1$. Using the open simulation protocol, the approximate state $|\tilde{\Psi}_1\rangle$ is computed by using $|0\rangle$ as initial state and C_1 as circuit. Similarly, the approximate state $|\tilde{\Psi}_2(x)\rangle$ is computed by using the open simulation protocol with $|x\rangle$ as initial circuit and $C_2^\dagger$ as circuit. Finally, the approximate amplitude $\tilde{a}_x$ is computed as:

$$\tilde{a}_x = \langle\Psi_2(x)|C_M|\Psi_1\rangle\,. \tag{H19}$$

As discussed in [34], $\tilde{a}_x$ can be seen as amplitudes extracted from a quantum state with a fidelity $F = F_1\overline{F_2}$, with F_1 and $\overline{F_2}$ being the fidelity of $|\tilde{\Psi}_1\rangle$ and the average fidelity of $|\tilde{\Psi}_2(x)\rangle$ respectively. Because both $|0\rangle$ and $|x\rangle$ are MPS with bond dimension 0, $F_1\overline{F_2}$ might be larger than the fidelity one obtains with the open simulation protocol. However, unlike the open simulation, multiple runs are needed to get the required number of bitstrings.

Fig. 18 shows the fidelity of amplitudes sampled using the close simulation protocol at fixed bond dimension $\chi = 8$, compared to the exact fidelity. Each point correspond to a different circuit, and circuits are split so that C_1 and C_2 contain 8 cycles while C_M contains 4 cycles. The average is performed over bitstrings.

4. XEB for approximate tensor representations

We now explain why, at sufficiently large depth, XEB is still a good estimator of fidelity for approximate tensor representations. The optimal approximate tensor representation Eq. (H3) is based on the Schmidt decomposition, that is

$$|\tilde{\psi}\rangle = \frac{1}{\sqrt{F}} \sum_{\alpha=1}^{\chi} S_\alpha \, |\nu_\alpha\rangle \ , \tag{H20}$$

where $|\nu_\alpha\rangle = |l_\alpha\rangle \, |r_\alpha\rangle$ and $F = \sum_{\alpha=1}^{\chi} S_\alpha^2$ is the fidelity.

We first show that in the limit of large depth the left and right singular vectors $\{|l_\alpha\rangle\}$ and $\{|r_\alpha\rangle\}$ are Haar random states. Note that these are the singular vectors of a matrix M with Gaussian entries (a Haar random matrix), see App. H 1. For any unitaries U and V we also have that $UMV^\dagger$ is a matrix with Gaussian entries. This implies that the distribution of singular vectors is invariant under unitary transformation, that is, the singular vectors are Haar random. Therefore, we approximate the singular vectors as having i.i.d Gaussian random real and imaginary parts.

We first give an explanation for why XEB is a good estimator of fidelity in this case, followed by a formal proof. We can write

$$|\psi\rangle = \sqrt{F} \, |\tilde{\psi}\rangle + \sqrt{1-F} \, |\perp\rangle \tag{H21}$$

where

$$|\perp\rangle = \frac{1}{\sqrt{1-F}} \sum_{\alpha=\chi+1}^{D_1} S_\alpha \, |\nu_\alpha\rangle \ . \tag{H22}$$

In the case of linear XEB, $f(p_j) = Dp_j$, we have

$$D \sum_j \tilde{p}_j p_j = F \sum_j D\tilde{p}_j^2 + (1-F) D \sum_j \tilde{p}_j \perp_j + \sqrt{F(1-F)} D \sum_j 2\mathrm{Re}(\langle j|\tilde{\psi}\rangle \langle \perp | j\rangle) \ , \tag{H23}$$

where p_j are the ideal probabilities $p_j = |\langle j|\psi\rangle|^2$ and $\tilde{p}_j = |\langle j|\tilde{\psi}\rangle|^2$. For $D_1 \gg \chi \gg 1$ both $|\tilde{\psi}\rangle$ and $|\perp\rangle$ converge to independent Haar random states. Therefore

$$D \sum_j \tilde{p}_j^2 \simeq 2 \tag{H24}$$

$$D \sum_j \tilde{p}_j \perp_j \simeq D^2 \langle\langle \tilde{p}_j \perp_j \rangle\rangle \simeq D^2 \langle\langle \tilde{p}_j \rangle\rangle \langle\langle \perp_j \rangle\rangle \simeq 1 \tag{H25}$$

$$2D \sum_j \mathrm{Re}(\langle j|\tilde{\psi}\rangle \langle \perp | j\rangle) \simeq 0 \ , \tag{H26}$$

where $\langle\langle\cdot\rangle\rangle$ denotes average over random states $|\psi\rangle$ (or circuits, see App. A). Therefore

$$D \sum_j \tilde{p}_j p_j \simeq F \ , \tag{H27}$$

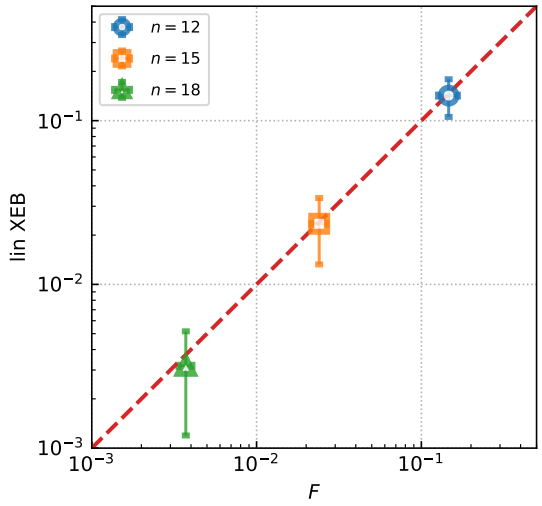


FIG. 19. The plot compares the exact fidelity F to the linear XEB for close simulations with a fixed bond dimension of $\chi = 8$. Circuits for the close simulations are split in three parts of $|C_1| = 8$, $|C_M| = 4$ and $|C_2| = 8$ cycles respectively. The error is obtained by averaging over multiple circuits. For all the instances, the pattern ABCDCDAB is used.

as we wanted.

We note that for very small χ, such as $\chi = 1$, linear XEB overestimates the fidelity. Indeed, in this case we have

$$D \sum_j \tilde{p}_j^2 \ = \ D \sum_{a,b} |\langle a|l_\alpha\rangle|^4 |\langle b|r_\alpha\rangle|^4 \ \simeq \ 4 \ . \tag{H28}$$

Nevertheless, this case is uninteresting for simulations, as $F \simeq \lambda_+^2/D_1$ corresponds to a very small fidelity. Numerical results for small quantum systems, Fig. 19, confirm the correspondence between exact fidelity and the XEB for approximate quantum states using close simulations.

We now give a detailed proof. Below till the end of this subsection, for the simplicity of notation, we will use $\mathbb{E}[\cdot]$ to denote the average over states $\langle\langle\cdot\rangle\rangle$.

We recall that for a $L \times L$ random $\beta-$Haar distributed unitary Q ($\beta = 1$ for real and $\beta = 2$ for complex and $\beta = 4$ for quaternion unitaries) with entries $q_{i,j}$ we have the following expectations (see Table IV in [38]):

expectation values
$\mathbb{E}[\lvert q_{i,j}\rvert^4] = \frac{\beta+2}{L(\beta L+2)}$
$\mathbb{E}[\lvert q_{i,j} q_{i,k}\rvert^2] = \frac{\beta}{L(\beta L+2)}$

We remind the reader that the exact and approximation outputs (from Eqs. (H3) and (H20)) are

$$|\psi\rangle := \sum_{\alpha=1}^{\sqrt{D}} S_\alpha |l_\alpha\rangle \otimes |r_\alpha\rangle := \sum_{\alpha=1}^{\sqrt{D}} S_\alpha |\nu_\alpha\rangle$$

$$|\tilde{\psi}\rangle := \frac{1}{\sqrt{F}} \sum_{\alpha=1}^{\chi} S_\alpha |l_\alpha\rangle \otimes |r_\alpha\rangle := \frac{1}{\sqrt{F}} \sum_{\alpha=1}^{\chi} S_\alpha |\nu_\alpha\rangle \, .$$

Let $p_j := |\langle j|\psi\rangle|^2$ and $\tilde{p}_j := |\langle j|\tilde{\psi}\rangle|^2$. We have $\langle j|\psi\rangle = \sum_{\alpha=1}^{\sqrt{D}} S_\alpha \langle j|\nu_\alpha\rangle$

$$p_j = \left|\sum_{\alpha=1}^{\sqrt{D}} S_\alpha \langle j|\nu_\alpha\rangle\right|^2 = \sum_{\alpha,\beta=1}^{\sqrt{D}} S_\alpha S_\beta \ \langle j|\nu_\alpha\rangle\langle\nu_\beta|j\rangle \ ,$$

$$\tilde{p}_j = \frac{1}{F}\left|\sum_{\alpha=1}^{\chi} S_\alpha \langle j|\nu_\alpha\rangle\right|^2 = \frac{1}{F}\sum_{\alpha,\beta=1}^{\chi} S_\alpha S_\beta \ \langle j|\nu_\alpha\rangle\langle\nu_\beta|j\rangle \ .$$

As stated above $\langle j|\nu_\alpha\rangle = \langle j|(|l_\alpha\rangle\otimes|r_\alpha\rangle) = v_\alpha w_\alpha$ is simply a product of two complex numbers, where v_α and w_α represent an entry of $|l_\alpha\rangle$ and $|r_\alpha\rangle$ respectively.

Lemma 1. *$D\mathbb{E}\left[\sum_j p_j\tilde{p}_j\right] - 1 \approx F + O(1/\sqrt{D})$, where the expectation is over the random haar vectors $|l_\alpha\rangle$ and $|r_\alpha\rangle$ and singular values S_α.*

Proof. We first compute $\mathbb{E}[p_j\tilde{p}_j]$. Treating singular values as independent from the vectors we have

$$\begin{aligned}\mathbb{E}[p_j\tilde{p}_j] =& \tfrac{1}{F}\textstyle\sum_{\alpha,\beta=1}^{\sqrt{D}}\sum_{c,d=1}^{\chi}\{\mathbb{E}[S_\alpha S_\beta S_c S_d]\times \\ & \mathbb{E}[\langle j|\nu_\alpha\rangle\langle\nu_\beta|j\rangle\langle j|\nu_c\rangle\langle\nu_d|j\rangle]\} \\ =& \tfrac{1}{F}\textstyle\sum_{\alpha,\beta=1}^{\sqrt{D}}\sum_{c,d=1}^{\chi}\mathbb{E}\{[S_\alpha S_\beta S_c S_d]\times \\ & \mathbb{E}[v_\alpha w_\alpha\overline{v_\beta w_\beta}v_c w_c\overline{v_d w_d}]\} \ . \end{aligned} \tag{H29}$$

First of all the vectors $|l_\alpha\rangle$ and $|r_\alpha\rangle$ are independent and power of their entries vanish over the complex field because of the invariance of Haar measure. Let

$$\begin{aligned}g_{j,\chi,D} :=& \tfrac{1}{F}\textstyle\sum_{\alpha,\beta=1}^{\sqrt{D}}\sum_{c,d=1}^{\chi}\{\mathbb{E}[S_\alpha S_\beta S_c S_d]\ \mathbb{E}[v_\alpha v_c\overline{v_\beta v_d}]\times \\ & \mathbb{E}[w_\alpha w_c\overline{w_\beta w_d}]\}. \end{aligned} \tag{H30}$$

The non-zero contributions in the sum are three cases:

$$\begin{aligned}&\text{case 1: } \alpha=\beta\neq c=d \\ &\text{case 2: } \alpha=d\neq c=\beta \\ &\text{case 3: } \alpha=d=c=\beta\end{aligned}$$

Before we indulge in computing these cases one by one, let us focus on the expectation with respect to the *entries* as the entries of a Haar unitary have correlations. When $\alpha=\beta\neq c=d$ and v_α and v_c do not belong to the same row of the unitary matrix induced by singular value decomposition whose columns are $|l_\alpha\rangle$, we have

$$\mathbb{E}[|v_\alpha^{(1)}|^2|v_c^{(2)}|^2] = \frac{1}{D}\sqrt{D}(\sqrt{D}-1)\frac{1}{D} = \frac{1}{D}(1-\frac{1}{\sqrt{D}}).$$

However, when $\alpha=\beta\neq c=d$ and v_α and v_c come from the *same* row or column of the unitary matrix, then from the second row of the Table above we have

$$\begin{aligned}\mathbb{E}[|v_\alpha^{(1)}|^2|v_c^{(1)}|^2] &= \frac{1}{D}\sqrt{D}\frac{\beta}{\sqrt{D}\left(\beta\sqrt{D}+1\right)} \\ &= \frac{1}{D}\frac{\beta}{\left(\beta\sqrt{D}+1\right)} \\ &= \frac{1}{D\sqrt{D}}\left(1-\frac{1}{\beta\sqrt{D}}\right) \ . \end{aligned} \tag{H31}$$

The latter is of lower order. We ignore the terms of lower order below and proceed to calculate the three cases by assuming that the entries do not come from the same row or column of the inducing random haar unitary matrix.

Recall from the asymptotic scaling of the purity Eq. (H16) that $\sum_{c=1}^{\chi}\mathbb{E}[S_c^4] \leq 2/\sqrt{D}$.

Case 1: $\alpha=\beta\neq c=d$. We have up to $O(1/\sqrt{D})$ from Eq. (H30)

$$\begin{aligned}g_{j,\chi,D} &= \frac{1}{F}\sum_{\alpha=1,\alpha\neq c}^{\sqrt{D}}\sum_{c=1}^{\chi}\mathbb{E}[S_\alpha^2 S_c^2]\ \mathbb{E}[|v_\alpha|^2|v_c|^2]\ \mathbb{E}[|w_\alpha|^2|w_c|^2] \\ &= \frac{1}{FD^2}\sum_{\alpha=1,\alpha\neq c}^{\sqrt{D}}\sum_{c=1}^{\chi}\mathbb{E}[S_\alpha^2 S_c^2] \\ &= \frac{1}{FD^2}\left(\sum_{\alpha=1}^{\sqrt{D}}\sum_{c=1}^{\chi}\mathbb{E}[S_\alpha^2 S_c^2] - \sum_{c=1}^{\chi}\mathbb{E}[S_c^4]\right) \\ &= \frac{1}{FD^2}\left(F\sum_{\alpha=1}^{\sqrt{D}}\mathbb{E}[S_\alpha^2] - \sum_{c=1}^{\chi}\mathbb{E}[S_c^4]\right) \\ &= \frac{1}{FD^2}\left(F\sum_{\alpha=1}^{\sqrt{D}}\mathbb{E}[S_\alpha^2] - O(1/\sqrt{D})\right) \\ &= \frac{1}{D^2}\left(1-O(1/\sqrt{D})\right). \end{aligned} \tag{H32}$$

Case 2: $\alpha=d\neq c=\beta$. We have up to $O(1/\sqrt{D})$ from Eq. (H30)

$$\begin{aligned}g_{j,\chi,D} &= \frac{1}{F}\sum_{\alpha=1,\alpha\neq c}^{\chi}\sum_{c=1}^{\chi}\mathbb{E}[S_\alpha^2 S_c^2]\ \mathbb{E}[|v_\alpha|^2|v_c|^2]\ \mathbb{E}[|w_\alpha|^2|w_c|^2] \\ &= \frac{1}{FD^2}\sum_{\alpha=1,\alpha\neq c}^{\chi}\sum_{c=1}^{\chi}\mathbb{E}[S_\alpha^2 S_c^2] \\ &= \frac{1}{FD^2}\left(\sum_{\alpha=1}^{\chi}\sum_{c=1}^{\chi}\mathbb{E}[S_\alpha^2 S_c^2] - \sum_{c=1}^{\chi}\mathbb{E}[S_c^4]\right) \\ &= \frac{1}{FD^2}\left(\sum_{\alpha=1}^{\chi}\sum_{c=1}^{\chi}\mathbb{E}[S_\alpha^2 S_c^2] - O(1/\sqrt{D})\right) \\ &= \frac{1}{FD^2}\left(F\sum_{\alpha=1}^{\chi}\mathbb{E}[S_c^2] - O(1/\sqrt{D})\right) \\ &= \frac{1}{D^2}\left(\sum_{\alpha=1}^{\chi}\mathbb{E}[S_\alpha^2] - O(1/\sqrt{D})\right) \\ &= \frac{F}{D^2}\left(1-O(1/\sqrt{D})\right). \end{aligned}$$

Case 3: $\alpha=\beta=c=d$. From Eq. (H30) and the first

row of the Table above, we have

$$\begin{aligned} g_{j,\chi,D} &= \frac{1}{F}\sum_{c=1}^{\chi}\mathbb{E}[S_c^4]\;\mathbb{E}[|v_c|^4]\mathbb{E}[|w_c|^4] \\ &= \frac{1}{F}\left(\frac{\beta+2}{2\sqrt{D}(\sqrt{D}+1)}\right)^2\sum_{c=1}^{\chi}\mathbb{E}[S_c^4] \\ &\approx \frac{4}{D^2}\frac{\sum_{c=1}^{\chi}\mathbb{E}[S_c^4]}{F}\left(1-\frac{2}{D^{1/2}}\right) \\ &= O(D^{-5/2}). \end{aligned} \tag{H33}$$

We can now compute the desired result. First of all from Eq. (H29) and the above 3 cases we have

$$\mathbb{E}\sum_j \tilde{p}_j p_j = \frac{1}{D}+\frac{F}{D}+O(D^{-3/2})\,.$$

Therefore,

$$D\mathbb{E}\left[\sum_j \tilde{p}_j p_j\right]-1 = F+O(1/\sqrt{D}), \tag{H34}$$

which proves our result. □

We now consider the special case of $\chi = 1$ where the approximate state $|\tilde{\psi}\rangle$ is taken to be a product state.

Corollary 1. *When $\chi = 1$, then $D\mathbb{E}\left[\sum_j \tilde{p}_j p_j\right] - 1 = 3F + O(F/\sqrt{D})$.*

Proof. In this case

$$\mathbb{E}[p_j\tilde{p}_j] = \sum_{\alpha,\beta=1}^{\sqrt{D}}\mathbb{E}[S_\alpha S_\beta]\;\mathbb{E}[v_\alpha w_\alpha \overline{v_\beta w_\beta}|v_1|^2|w_1|^2] \tag{H35}$$

The only non-zero contributions come from $\alpha = \beta = 1$ and $\alpha, \beta \geq 2$. We have

$$\begin{aligned} \mathbb{E}[p_j\tilde{p}_j] &= \mathbb{E}[S_1^2]\;\mathbb{E}[|v_1|^4|w_1|^4] \\ &\quad + \sum_{\alpha,\beta=2}^{\sqrt{D}}\mathbb{E}[S_\alpha S_\beta]\;\mathbb{E}[v_\alpha w_\alpha\overline{v_\beta w_\beta}|v_1|^2|w_1|^2] \\ &= F\;\mathbb{E}[|v_1|^4]\mathbb{E}[|w_1|^4] \\ &\quad + \sum_{\alpha=2}^{\sqrt{D}}\mathbb{E}[S_\alpha^2]\;\mathbb{E}[|v_\alpha|^2]\mathbb{E}[|w_\alpha|^2]\mathbb{E}[|v_1|^2]\mathbb{E}[|w_1|^2] \\ &= F\;\mathbb{E}[|v_1|^4]\mathbb{E}[|w_1|^4] + \frac{1}{D^2}(1-F) \\ &= F\;\left(\frac{\beta+2}{2\sqrt{D}(\sqrt{D}+1)}\right)^2 + \frac{1}{D^2}(1-F) \\ &= F\;\frac{4}{D^2}\frac{1}{\left(1+1/\sqrt{D}\right)^2} + \frac{1}{D^2}(1-F) \\ &= F\;\frac{4}{D^2}\left(1-2/\sqrt{D}\right) + \frac{1}{D^2}(1-F) \\ &= \frac{3F}{D^2}+\frac{1}{D^2}+O(F\;D^{-5/2}). \end{aligned}$$

We conclude the desired final result

$$D\sum_j \mathbb{E}[p_j\tilde{p}_j] = 3F+1+O(F\;D^{-1/2}).$$

□

5. Quantifying entanglement with Clifford circuits

We derived an analytical and numerical bound on the fidelity of the tensor product approximation, Eqs. (H14) and (H15), which depend on the reduced purity. We now study the reduced purity growth rate. The average Pauli error between the fsim gates used in the experiment and the Clifford gate iSWAP^{-1} is $\sim 1\%$. For the purposes of quantifying reduced purity growth we therefore approximate fsim gates with iSWAP^{-1}. The one-qubit gates are $Z^pX^{1/2}Z^{-p}$ with $p \in \{-1, -1/4, -1/2, \ldots, 3/4\}$ [39]. We can study a related ensemble of Clifford circuits by reducing the parameter p of the one-qubit gates to the set $p \in \{-1, -1/2, 0, 1/2\}$. Note that the reduced purity produced by Clifford circuits is efficient to calculate numerically [40].

Consider the average purity of the reduced state $\langle\langle \operatorname{tr}\rho_L^2\rangle\rangle$, where ρ_L is the partial trace of $|\psi\rangle$ on the left qubits. We now show that layer by layer this average is the same for the random circuits and the corresponding Clifford circuits of the previous paragraph. One intuition why this might be true is that Clifford circuits are a two-design. Nevertheless, we are interesting in the growth rate, not the average over Clifford circuits. Therefore, we use a different technique.

First note that $\operatorname{tr}\rho_L^2$ can be written as

$$\sum_{z_L}\langle z_L|\sum_{z_R}\langle z_R|\,\rho\,|z_R\rangle\,|z_L\rangle\otimes\langle z_L|\sum_{z_R}\langle z_R|\,\rho\,|z_R\rangle\,|z_L\rangle\;,$$

where $\{z_L\}$ ($\{z_R\}$) is a basis for the left (right) patch. Therefore, this quantity can be calculated from two replicas of the output state of the circuit of interest $\rho\otimes\rho$. As shown in SM D and Ref. [13], for the circuits of interest, the average of an observable with two replicas can be calculated with a Markov chain describing the evolution of the density matrix in the basis of Pauli strings. If we denote the basis of normalized Pauli strings as $\{s\}$ we can write any operator as $\rho = \sum_s \operatorname{tr}(\rho s)s$. It follows that

$$U\rho U^\dagger = \sum_s \operatorname{tr}(\rho s)\sum_{s'}\operatorname{tr}(s'Us\,U^\dagger)s'\;. \tag{H36}$$

Using this relation repeatedly we reduce the evolution of a circuit to the evolution of gates over Pauli strings. Furthermore, the average over random circuits is composed of averages over the corresponding set $\{g\}$ of random gates. In the case of the evolution of two replicas the elementary step is

$$\mathbb{E}_g[\sigma^\alpha\otimes\sigma^{\alpha'}] = \tag{H37}$$
$$\sum_{\beta,\beta'}\frac{1}{|g|}\sum_g \operatorname{tr}(\sigma^\beta\otimes\sigma^{\beta'}g\otimes g\,\sigma^\alpha\otimes\sigma^{\alpha'}\,g^\dagger\otimes g^\dagger)\,\sigma^\beta\otimes\sigma^{\beta'}$$

where the tensor product is between the two replicas, and each gate is the same for both replicas. The initial state of each qubit with two replicas is

$$|0\rangle\langle 0|\otimes|0\rangle\langle 0| = \frac{1+Z}{2}\otimes\frac{1+Z}{2}\,. \tag{H38}$$

We obtain, both the Clifford and non-Clifford single-qubit gates above,

$$\mathbb{E}[II] = II \tag{H39}$$
$$\mathbb{E}[IZ] = \mathbb{E}[ZI] = 0 \tag{H40}$$
$$\mathbb{E}[ZZ] = \frac{1}{2}(XX+YY) \equiv \perp \tag{H41}$$

where we also define the symbol $\perp$. For shorthand, we also introduce the notation $\mathbb{Z} = ZZ$ and $\mathbb{I} = II$. We also obtain, again for both the Clifford and non-Clifford gates,

$$\mathbb{E}[\perp] = \frac{1}{2}(\mathbb{Z}+\perp)\,. \tag{H42}$$

Finally the transformation from applying iSWAP^{-1} to two qubits in each replica gives

$$\mathbb{I}\mathbb{Z} \to \mathbb{Z}\mathbb{I} \tag{H43}$$
$$\mathbb{I}\perp \to \perp\mathbb{Z} \tag{H44}$$
$$\mathbb{Z}\mathbb{Z} \to \mathbb{Z}\mathbb{Z} \tag{H45}$$
$$\perp\perp \to \perp\perp \tag{H46}$$
$$\perp\mathbb{Z} \to \mathbb{I}\perp \tag{H47}$$

Therefore the average purity of the reduced state $\langle\langle\rho_L\rangle\rangle$ is the same for both ensembles, as the transformations for the initial state of interest is the same for the average of Clifford and non-Clifford gates.

Entanglement is typically measured with the von Neumann entropy $-\rho_L \operatorname{tr}\rho_L$. Using Jensen's inequality this can be bounded with the Rényi entropy

$$-\operatorname{tr}\rho_L\log_2\rho_L \geq -\log_2\operatorname{tr}\rho_L^2\,. \tag{H48}$$

The average Rényi entropy can be bounded with purity of the reduced states using Jensen's inequality as

$$-\langle\langle\log_2\operatorname{tr}\rho_L^2\rangle\rangle \geq -\log_2\langle\langle\operatorname{tr}\rho_L^2\rangle\rangle\,. \tag{H49}$$

Figure 20 shows the reduced purity, as a function of the number of cycles, for different circuit sizes and cuts. Dashed lines are the reduced purity limit values, see Eq. (H50). For Sycamore-70 (this work), the reduced purity is close to its limit value at depth 10.

6. Reduced purity and distribution of singular values

We will now show numerically that the reduced purity is a good witness for the distribution of singular values, which undergoes through a sharp transition to its limiting

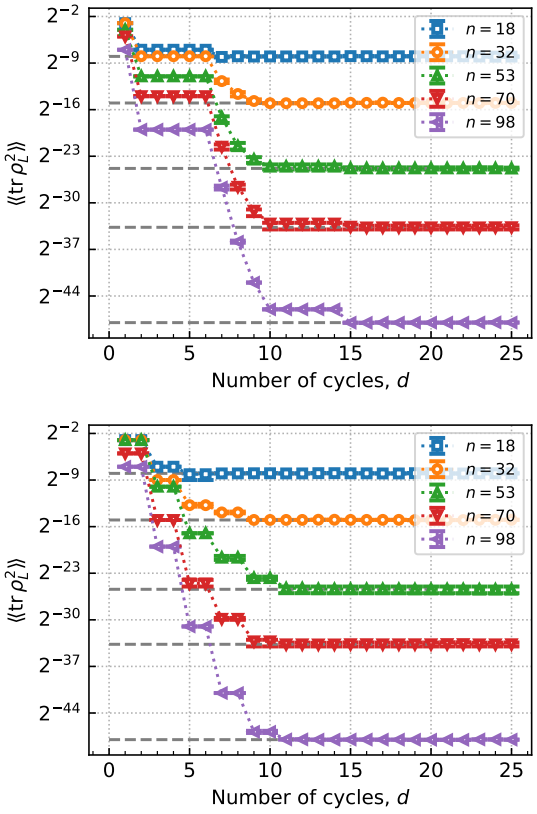


FIG. 20. The plots show the reduced purity as a function of the number of cycles, for different circuit sizes and cuts (diagonal cut for the top figure, and vertical cut for the bottom figure), using Clifford gates only. The dashed lines correspond to the reduced purity $\overline{\mathrm{pur}}$ in the random Haar limit. see Eq. (H50). For all the instances, the pattern ABCDCDAB is used.

value. We first need to understand what is the expected reduced purity and the corresponding standard deviation for a Haar random state. In Sec. H 1 we gave the distribution of the normalized singular values $s = \sqrt{D_1}S$, where D_1 is the small Hilbert space dimension (between the halves in which the state is being divided). The expected purity is

$$\overline{\mathrm{pur}} = \left\langle\!\!\left\langle \sum_{\alpha=1}^{D_1} S_\alpha^4 \right\rangle\!\!\right\rangle \simeq \frac{1}{D_1^2}\left\langle\!\!\left\langle \sum_{\alpha=1}^{D_1} s_\alpha^4 \right\rangle\!\!\right\rangle = \frac{1}{D_1}\langle\langle s_\alpha^4\rangle\rangle\,. \tag{H50}$$

The variance is

$$\begin{aligned}\mathrm{Var}(\mathrm{pur}) &= \mathrm{Var}\left(\sum_{\alpha=1}^{D_1} S_\alpha^4\right) \simeq \frac{1}{D_1^4}\sum_{\alpha=1}^{D_1}\mathrm{Var}(s_\alpha^4) \\ &= \frac{1}{D_1^3}\mathrm{Var}(s_\alpha^4)\,.\end{aligned} \tag{H51}$$

Therefore, we can say that the reduced purity has converged to its limiting value at a given depth when

$$\frac{\operatorname{tr}\rho_L^2 - \overline{\text{pur}}}{\sqrt{\text{Var(pur)}}} \in O(1) \tag{H52}$$

Note that the variance decreases exponentially.

Figure 21-Top shows the distance of the reduced purity in units of the standard deviation as a function of the number of cycles. As expected, the depth for which the reduced purity is exponentially close to its limit value increases with the system size. Figure 21-Bottom shows instead how the Kolmogorov-Smirnov $p-$value between the singular values S_α and the Haar random distribution of singular values Eq. (H10) transitions when the purity reaches its limiting value in units of standard deviation. As one can see, there is a sharp transition so that only once the reduced purity is appropriately close to its limit value, the distribution of S_α truly follows the distribution of singular values in the random Haar limit.

7. Bounding the approximate tensor representation performance for close simulations

Using Eqs. (H14) and (H15), it is possible to lower bound the required χ to achieve a target fidelity F for close simulations (see App. H 3). For our bounds, we split circuits C of m cycles in three parts of $|C_1| = m-2$, $|C_M| = 4$ and $|C_2| = m-2$ cycles respectively. Moreover, we assume that it is possible to compute $|\tilde{\Psi}_1\rangle$ and $|\tilde{\Psi}_2(x)\rangle$ with a single truncation each. While being unrealistic for any practical purpose, it allows us to find analytical and semi-analytical bounds since every realistic simulation would require more than one truncation.

Recalling that the final fidelity of a close simulation is $F = F_1\overline{F_2}$, with F_1 and $\overline{F_2}$ being the fidelity of $|\tilde{\Psi}_1\rangle$ and the average fidelity $|\tilde{\Psi}_2(x)\rangle$ respectively, it is possible to get an estimate of the optimal bond dimension χ for a given target fidelity F as:

$$\chi_{\text{an}} = \frac{F}{\langle\langle \operatorname{tr}\rho_L^2\rangle\rangle}, \tag{H53a}$$

$$\chi_{\text{nm}} = \frac{\mathcal{F}_\lambda^{-1}(\sqrt{F})}{\langle\langle \operatorname{tr}\rho_L^2\rangle\rangle} \geq \frac{\sqrt{F}}{\lambda_+^2\,\langle\langle \operatorname{tr}\rho_L^2\rangle\rangle}, \tag{H53b}$$

with $\lambda_+^2 \leq 2$ being the largest singular value, and χ_{an} and χ_{nm} being respectively the estimate for the bond dimension using either the analytical or the numerical bound. It is important to stress that the upper bounds provided by Eqs. (H53) are valid for arbitrary depths and bond dimensions χ, even if the quantum state has not yet reached the Porter-Thomas limit. For small target fidelity F, the ratio between χ_{an} and χ_{nm} becomes:

$$\frac{\chi_{\text{an}}}{\chi_{\text{nm}}} \approx \lambda_+^2\sqrt{F}. \tag{H54}$$

For a cut that split the qubits in two equal halves ($\lambda_+ = 2$), and for a target fidelity of $F = 10^{-4}$, one gets

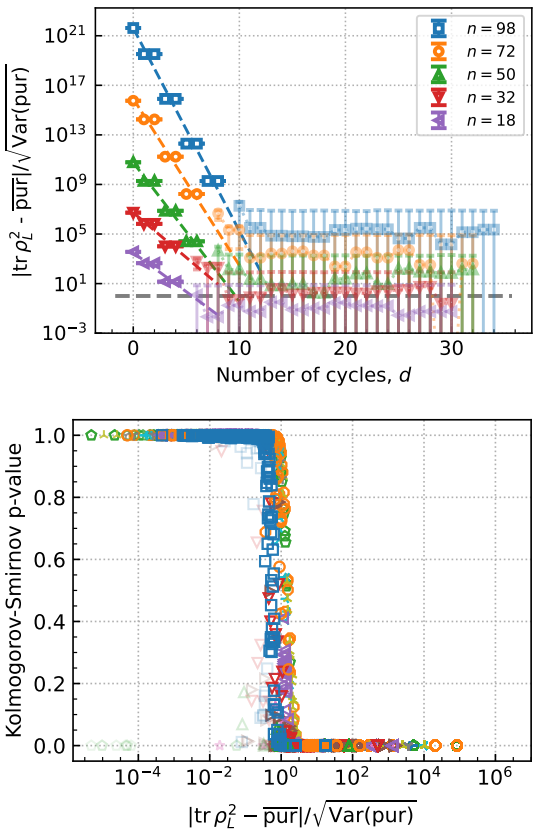


FIG. 21. (Top) Distance of the reduced purity from its limit value in units of the standard deviation as a function of depths, using Clifford gates only. The dashed line corresponds to 1. (Bottom) Kolmogorov-Smirnov $p-$value between the singular values S_α and the distribution of singular values for large depth, Eq. (H10), as a function of the distance between the reduced purity and its limit value in units of standard deviation. Each point corresponds to a different circuits (with the number of qubits ranging from $n = 8$ and $n = 24$) at given fixed depth. Lighter points correspond to datapoints outside the 90% two sided confidence interval. For all the instances, the pattern ABCDCDAB is used, and qubits are partitioned in two equal halves using a diagonal cut.

$\chi_{\text{nm}} \approx 25\,\chi_{\text{an}}$, that is the numerical bound is only 25 times larger than the analytical bound. This is consistent with what we observe in Fig. 22. Note the required FLOPs scale as $O(2^n\chi)$ if we represent the state with two equal size tensors.

Appendix I: Client-certified randomness generation with RCS

Randomness is a valuable resource with many applications and is a key resource in much of modern cryptography. In classical physics, the outcome of any experiment can in principle be determined from the initial conditions, so there is no such thing as true randomness. On

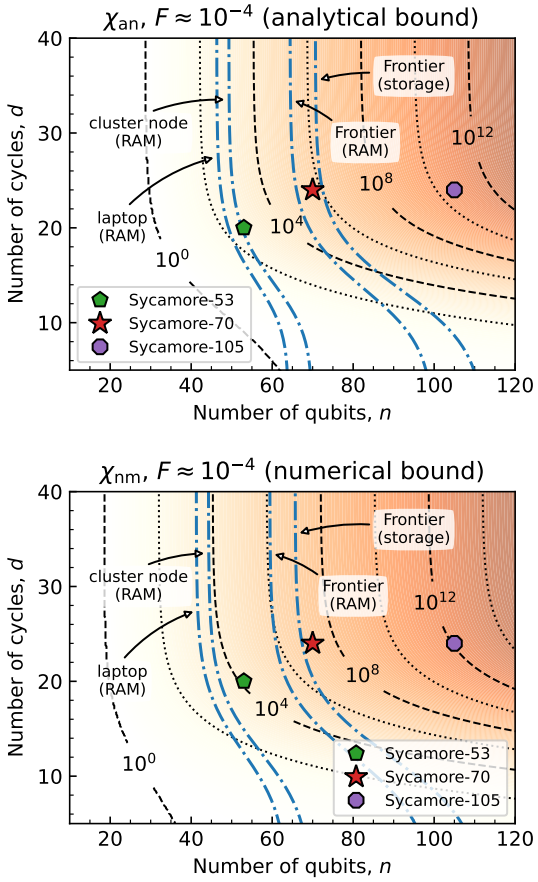


FIG. 22. Analytical (Top) and numerical (Bottom) upper bounds for the required bond dimension to achieve a target fidelity of $F = 10^{-4}$, by varying the number of qubits and depths. The memory footprint (blue dashed-and-dotted lines) is computed as the amount of memory required to store two `complex32` tensors of dimension $2^{n/2} \times \chi$. Laptop (RAM) = 32GB, cluster node (RAM) = 256GB, Frontier (RAM) = 9.2PB and Frontier (storage) = 700PB. The density map is obtained by averaging circuits with patterns ABCDCDAB/BADCDCBA and with diagonal/vertical cuts.

the other hand, quantum physical systems exhibit true randomness. The outcome of certain quantum processes is inherently random, meaning that no amount of prior information is sufficient to predict the outcome.

Client-certified randomness generation has been proposed as a potential application of RCS [41–43]. The proposed protocol works as follows. The client generates challenge random quantum circuits which she sends to an untrusted server that operates a quantum processor. The server responds to each challenge by sending a requested number of bitstrings within a given short amount of time. When the server is honest, it produces the bitstrings by sampling from the circuit using the quantum processor, so the bitstrings contain entropy due to the inherent randomness of quantum measurements. The client can then pass the raw bitstrings through a randomness extractor to obtain random bits of higher quality, in the sense of being closer to the uniform distribution.

The client is able to gain confidence that the returned sample of bitstrings is consistent with executing the challenge circuits by performing statistical tests, such as XEB. For cryptographic certification of randomness, these tests need to account for the possibility that the quantum operator is adversarial. Such an adversary will try to construct a set of bitstrings that pass the statistical tests despite having low or no entropy. Challenge circuits must be executed with fidelity greater than an agreed value F. In principle, the client allots a sampling time sufficiently shorter than the necessary time to simulate the same sampling (number of bitstrings and fidelity) by all known classical algorithms using reasonable computing resources. This way, the client can gain confidence that the bitstrings were indeed obtained using a quantum computer, and therefore are a source of quantum randomness. Unfortunately, in this protocol, the client needs to perform classical simulations if she wishes to certify the quantum randomness using XEB. However, the client can do this simulation a posteriori, running classical computations for a much longer time. Furthermore, the client can in principle use a large number of challenge circuits and select only a subset of them for verification. This way the client can force an adversarial server to perform expensive simulations on a large number of circuits while only expending computing resources verifying a much smaller subset.

There exist a tension between the need for practical verification, which incurs an exponential cost in this proposal, and the requirement that an adversary could not pass the same test deterministically. Furthermore, "spoofing" is typically a factor of F cheaper than the verification [19]. The 70 qubit circuits presented in this work are currently too big to be verified with XEB. At the same time, the computational cost of classical algorithms keeps improving (see SM G), as well as the performance of implementations in specialized hardware [44]. In this work we do not resolve this tension, and we leave open the problem of finding an efficient verification protocol, perhaps along the line of cryptographically secure proposals [45, 46], or more near-term obfuscation techniques [47]. We nevertheless study how this protocol could work if this problem is resolved or if a client is willing to expend sufficient compute resources to gain enough confidence against a potential deterministic adversarial server.

We summarize how our study of certified randomness compares to the original proposal introduced in Refs. [41, 43]. The original reference is more rigorous, under the assumption that the total system fidelity remains constant as the number of qubits scales, which requires quantum error correction [48]. Our proposal is less rigorous but more practical. First, we argue that the amount of randomness per circuit should scale as the number of samples times the system fidelity, not as the number of qubits as in Refs. [41, 43]. Second, we intro-

duce new statistical tests to detect adversarial sampling with lower randomness. Third, we implement an optimized (quantum-proof) Trevisan's randomness extractor [43, 49], which is important for a potential practical application given that the required input sizes are several orders of magnitude larger than those considered by previous implementations. We also propose a faster but less rigorous randomness extractor.

1. Entropy estimation

The output of the protocol is produced by applying a randomness extractor to the output of the quantum computer; see Sec. I 4. A randomness extractor takes an input from the source with a given min-entropy, together with a uniformly random seed, and it outputs a near-uniformly random bitstring of length proportional to the input min-entropy. The min-entropy of a random variable X is defined as minus the log of the maximum probability:

$$\text{min-entropy} = -\max_x \log_2 \left(\Pr[X = x]\right). \tag{I1}$$

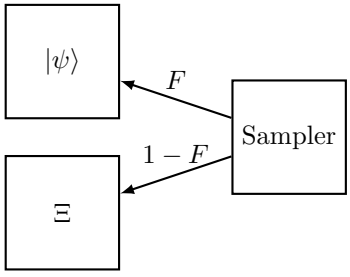
Below, we give bounds for the quantum min-entropy (see also Ref. [50]).

a. Entropy estimation for an honest server

The experimental output of a noisy quantum random circuit can be described by (see SM A)

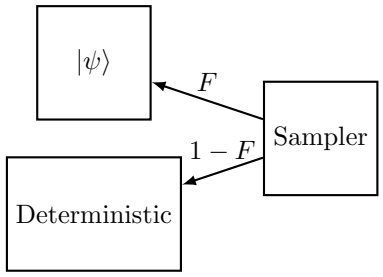
$$F\,|\psi\rangle\langle\psi| + (1-F)\Xi\,, \tag{I2}$$

where F is the experimental fidelity (probability of no error), $|\psi\rangle$ is the ideal output of the quantum circuit, and Ξ has trace one and is the result of errors. Measurement of this state can be interpreted as measuring the ideal quantum state $|\psi\rangle$ with probability F, and measuring the operator Ξ with probability $1-F$. This is depicted in the following diagram:



(I3)

For the purposes of quantum randomness generation, we take an adversarial approach with respect to the noise operator Ξ and consider it to be deterministic. The reason is that we are purely interested in the quantum entropy that is generated experimentally, and not in using potential "noise" or "errors" in the experiment as a source of entropy. Arguably, if we were willing to accept an entropy source based on "noise", there are simpler setups that do not require the use of a quantum processor. Furthermore, the potential entropy coming from the noise cannot be certified. Therefore, we model the sampling as depicted in the next diagram:



(I4)

Given the adversarial model above, the bitstring with the highest probability is the deterministic noise with probability $1-F$. Therefore for a sample of size k the min-entropy is

$$\text{min-entropy} = -\log_2\left((1-F)^k\right) \approx kF \tag{I5}$$

It is possible to obtain a tighter bound by ignoring the very unlikely event that all outputs of the experiment correspond to the "noise" term, which we are treating as deterministic. The approach is to use the ε-min-entropy or smooth min-entropy, that is, we bound the min-entropy ignoring events with cumulative probability smaller than some suitable small ε.

In the simplified model given by (I4) within a sample of size k the expected number of bitstrings obtained from the ideal output $|\psi\rangle$ is kF with the variance $kF(1-F)$. We wish to bound the probability of obtaining sufficiently many bitstrings from the ideal output. We now estimate this probability for obtaining a sequence of bitstrings from the ideal distribution up to a cumulative probability of $1-\varepsilon$. Assuming that we are sampling from a quantum processor with experimental fidelity F, then we can choose a constant c_1 and upper-bound the probability of obtaining at least

$$q := kF - c_1\sqrt{kF(1-F)} \tag{I6}$$

bitstrings from the *ideal* distribution, where c_1 is the number of standard deviations below the mean. Since the sample size is large enough (order of millions) we can use gaussianity where by $c_1 = 5$ implies that $\epsilon = 1.5\times10^{-12}$ rendering the probability of success $1 - 1.5\times10^{-12}$.

Suppose that the distribution of the ideal output probabilities $p(s) = |\langle s|\psi\rangle|^2$ follows the Porter-Thomas distribution with the probability density function

$$f(x) = e^{-x}, \tag{I7}$$

where $x = Dp(s)$ is the ideal bitstring probability scaled by the Hilbert space dimension $D = 2^n$. The average of minus log of the probability of one bitstring from the

Porter-Thomas distribution is (see Eq. (A7))

$$-D^2 \int_0^\infty \log(p) e^{-Dp} p \, dp = \log(D) - 1 + \gamma \,, \tag{I8}$$

where γ is Euler's constant. Then minus the log of the probability of a sequence of q independent ideal bitstrings is, by central limit theorem, a normal distribution with average $q(\log(D)-1+\gamma)$. The variance is upper-bounded by $q\pi^2/6$. Similar as above, we can choose a constant c_2 such that with high probability the ε-min-entropy is $q(\log(D)-1+\gamma)-c_2\sqrt{q\pi^2/6}$. Putting it all together, we obtain the following lower-bound for the ε-min-entropy

$$q(\log(D) - 1 + \gamma) - c_2 \sqrt{\frac{q\pi^2}{6}}. \tag{I9}$$

b. *Correction to the min-entropy*

The bound of Eq. (I9) represents the pure quantum min-entropy obtained from sampling a random quantum circuit. We now consider the situation in which a client has only black-box access (say via the cloud) to the quantum processor held by a server. We discuss deviations from the scenario of the previous section due to potential adversarial actions of the server, while still assuming that the server calls a quantum processor to obtain the output bitstrings.

In the previous section we bounded the number q of bitstrings obtained in a sample of size k using a quantum processor of fidelity F. An adversarial quantum server might be able to oversample sq bitstrings with $s \geq 1$ in the allotted time from the ideal quantum state before returning q bistrings. The server can also rearrange the bistrings in any predetermined way before returning them to the client. These operations lower the min-entropy and do not necessarily affect statistical tests such as the cross entropy. In order to bound the min-entropy of this multiset oversampling, we consider first a simplified model where the server samples from a uniform distribution of size D instead of the ideal quantum state.

We now give an expression for the probability of obtaining a given multiset S of size q when sampling sq times from the uniform distribution of D values. We can assume $q \ll D$ and that all the values in the set S are distinct. Let A_i denote the set of all sequences missing value i. We have

$$P(S) = 1 - \frac{\left|\bigcup_{i\in S} A_i\right|}{D^{sq}} \,. \tag{I10}$$

Note that

$$\left|\bigcap_{i\in I} A_i\right| = (D - |I|)^{sq} \,. \tag{I11}$$

Therefore, by the inclusion-exclusion principle, we have

$$\left|\bigcup_{i\in S} A_i\right| = \sum_{j=1}^{q} (-1)^{j-1} \binom{q}{j} (D-j)^{sq} \,, \tag{I12}$$

and

$$P(S) = 1 - \sum_{j=1}^{q} (-1)^{j-1} \binom{q}{j} \left(\frac{D-j}{D}\right)^{sq} \,. \tag{I13}$$

Although this expression is exact, note that all the terms have the same order in $1/D$, so it does not result in a compact estimate of the probability in the case of interest, $D \to \infty$.

We can also write different upper and lower bounds for $P(S)$. Let α denote a set of q indexes and B_α denote the set of words with the q given values in positions α. We first have

$$P(S) \leq \frac{\sum_\alpha |B_\alpha|}{D^{sq}} \,. \tag{I14}$$

There are $q!$ ways in which the q values can appear in the α positions, and D^{sq-q} possible choices for the other $sq-q$ positions. Therefore

$$|B_\alpha| = q! D^{sq-q} \,. \tag{I15}$$

There are are sq chose q ways to choose α. Therefore

$$P(S) \leq \binom{sq}{q} \frac{q!}{D^q} = \frac{(sq)_q}{D^q} \,, \tag{I16}$$

where

$$(sq)_q = \Pi_{j=0}^{q-1} (sq - j) \tag{I17}$$

This gives the following bound for the min-entropy

$$\text{min-entropy} \geq q \log D - \log(sq)_q \,. \tag{I18}$$

We can obtain a related lower bound by considering the sets C_α including words with the q values of interest in positions α, and none of those values anywhere else. Then

$$P(S) \geq \frac{\sum_\alpha |C_\alpha|}{D^{sq}} \tag{I19}$$

$$= \binom{sq}{q} \frac{q!(D-q)^{sq-q}}{D^{sq}} \tag{I20}$$

$$= \frac{(sq)_q}{D^q} \left(1 - \frac{q}{D}\right)^{q(s-1)} \,. \tag{I21}$$

Therefore, in the limit of $D \to \infty$, we have

$$P(S) = \frac{(sq)_q}{D^q} \left(1 - O\left(\frac{q^2 s}{D}\right)\right) \tag{I22}$$

$$\text{min-entropy} = q \log D - \log(sq)_q + O\left(\frac{q^2 s}{D}\right) \tag{I23}$$

$$\simeq q \log D - q \log sq + q \,. \tag{I24}$$

We have seen that multiset sampling can lower the min-entropy by a factor $\log((sq)_q)$. Applying this to the honest server min-entropy bound, Eq. (I9), gives a bound for the multiset sampling min-entropy

$$q(\log(D) - 1 + \gamma) - c_2 \sqrt{\frac{q\pi^2}{6}} - \log((sq)_q) \,. \tag{I25}$$

2. Repeated bitstrings

In the previous section we ignored the possibility of repeated bitstrings in the adversarial server sampling. We study this now. We denote the total sampling budget of the adversarial server (sq in the previous section) by β. We will see that the client can require the server to return unique bitstrings, and this has little effect in the linear XEB as long as $\beta \ll D$. An adversarial server can also postselect to bistrings that appear at least twice to artificially boost the nominal "fidelity" as measured by XEB. We will see that this effect is negligible as long as $s \ll \sqrt{D/q}$.

a. *Probabilities for repeated bitstrings*

The probability that a bitstring j appears exactly c times is

$$p'_j(c) = \binom{\beta}{c} p_j^c (1-p_j)^{\beta-c} \,. \tag{I26}$$

For large enough sampling budget β, there may occur collisions, i.e., repeated strings. We can calculate the expected number M of strings appearing with each multiplicity c, and the corresponding ideal probability value A. In the following, we derive closed formulas up to first-order approximation, confirming the formulas (I44), (I50) conjectured in Ref. [50, App. D].

Lemma 1. *Assuming $D+\beta \gg c$, the expected number of bitstrings that appear exactly c times is*

$$M_{\beta,c} = \binom{\beta}{c} \frac{D^{1-c} c!}{\left(1+\frac{\beta}{D}\right)^{c+1}} \left(1 + O\left(\sqrt{\frac{(2c)!}{D}}\right)\right) \,. \tag{I27}$$

Proof. The expected number of bitstrings that appear exactly c times is

$$M_{\beta,c} = \sum_j p'_j(c) \,. \tag{I28}$$

First note that

$$M_{\beta,c} = \sum_j p'_j(c) \tag{I29}$$

$$= D \left\langle\langle p'(c) \right\rangle\rangle + O\left(\sqrt{D \operatorname{Var}(p'(c))}\right) \,, \tag{I30}$$

where, as in App. A, we use the approximation that for large D the probabilities $p'_j(c)$ are i.i.d.

We can write

$$\left\langle\langle p'(c) \right\rangle\rangle = \binom{\beta}{c} I(\beta, c) \,, \tag{I31}$$

where $I(\beta, c)$ is the expectation value of $p^c(1-p)^{\beta-c}$. This can be calculated as

$$I(\beta,c) = \int_0^1 p^c (1-p)^{\beta-c} dF(p) \tag{I32}$$

$$= \int_0^1 p^c (1-p)^{\beta-c} (D-1)(1-p)^{D-2} dp \tag{I33}$$

$$= (D-1)c! \frac{(D+\beta-c-2)!}{(D+\beta-1)!} \tag{I34}$$

$$= (D-1)c! \frac{1}{(D+\beta-1)_{c+1}} \,, \tag{I35}$$

where the last expression uses a falling factorial in the denominator. Therefore

$$\left\langle\langle p'(c) \right\rangle\rangle = \binom{\beta}{c} \frac{(D-1)c!}{(D+\beta-1)_{c+1}} \,, \tag{I36}$$

We are interested in the value for large D, so we can use the approximation

$$\left\langle\langle p'(c) \right\rangle\rangle \simeq \binom{\beta}{c} \frac{(D-1)c!}{\left(D+\beta-1-\frac{c}{2}\right)^{c+1}} \,. \tag{I37}$$

This approximation is valid for

$$D + \beta \gg c \,, \tag{I38}$$

which is always the case in the regime of parameters we are interesting in.

We can also estimate the variance

$$\frac{\operatorname{Var}(p'(c))}{\binom{\beta}{c}^2} \tag{I39}$$

$$= \frac{(D-1)(2c)!}{(D+2\beta-1-c)^{2c+1}} - \left(\frac{(D-1)c!}{\left(D+\beta-1-\frac{c}{2}\right)^{c+1}}\right)^2 \tag{I40}$$

$$= D^{-2c} \frac{(1-\frac{1}{D})(2c)!}{\left(1+2\frac{\beta}{D}-\frac{1}{D}-\frac{c}{D}\right)^{2c+1}} - D^{-2c}\left(\frac{(1-\frac{1}{D})c!}{\left(1+\frac{\beta}{D}-\frac{1}{D}-\frac{c}{2D}\right)^{c+1}}\right)^2 \,. \tag{I41}$$

Ignoring small terms we get

$$\operatorname{Var}(p'(c)) = \binom{\beta}{c}^2 D^{-2c} \left((2c)! - (c!)^2 + O(\beta/D)\right) \,. \tag{I42}$$

Keeping only the dominant term $(2c)!$ completes the proof of the lemma.

Note that we also ignore terms $O(1/D)$ and $O(c/D)$ for consistency with the fluctuations from the variance. □

In Eq. (I27) we can use $\beta \gg c$ to write

$$\binom{\beta}{c} \simeq \frac{\left(\beta - \frac{c}{2}\right)^c}{c!} \,. \tag{I43}$$

Plugging this back and ignoring again terms $O(1/D)$ and $O(c/D)$ we get

$$M_{b,c} \simeq D \frac{b^c}{(1+b)^{c+1}} \,, \tag{I44}$$

where $b = \beta/D$.

We are also interested in the expected value of the simulated or ideal probability for the bistrings that are obtained exactly c times in a β-sample.

Lemma 2. *The expected value of the ideal probability for the bistrings that are obtained exactly c times is*

$$A_{\beta,c} = \frac{1}{D} \frac{c+1}{1+b} \,. \tag{I45}$$

Proof. The probabilities of bitstrings conditioned on appearing exactly c times are proportional to $p'_j(c)$, normalized so that their sum is 1. That is, the conditional probabilities are $p'_j(c)/M_{b,c}$. Therefore, the expected value of the simulated probability conditioned on appearing c times has the expression

$$A_{\beta,c} = \frac{1}{M_{\beta,c}} \sum_j p'_j(c)\, p_j \,. \tag{I46}$$

Using the same methodology as in Lemma 1 we have

$$A_{\beta,c} = \frac{\int_0^1 p^{c+1}(1-p)^{\beta-c} dF(p)}{\int_0^1 p^c (1-p)^{\beta-c} dF(p)} \tag{I47}$$

$$= \frac{I(\beta+1, c+1)}{I(\beta, c)} \tag{I48}$$

$$= \frac{(c+1)!}{c!} \frac{(D+\beta-1)_{c+1}}{(D+\beta)_{c+2}} \tag{I49}$$

$$= \frac{1}{D} \frac{c+1}{1+b} \,. \tag{I50}$$

□

The expected number of unique bitsrings in a β-sample follows from Eq. (I44) and is given by the expression

$$M_\beta = \sum_{c=1}^{\infty} M_{\beta,c} = D \frac{\beta}{D+\beta} \,. \tag{I51}$$

b. Linear cross-entropy with unique bitstrings

Following the same logic as in Eq. (I46), we can calculate the expected value of the linear cross entropy when an honest server returns unique bitstrings. The probabilities of bitstrings conditioned on appearing at least one time are proportional to

$$p''_j(c) = \sum_{c=1}^{\infty} p'_j(c) \,, \tag{I52}$$

normalized so that their sum is 1. The corresponding linear cross entropy is

$$D \sum_j \frac{p''_j(c)}{M_\beta} p_j - 1 = \frac{D}{M_\beta} \sum_{c=1}^{\infty} A_{\beta,c} M_{\beta,c} - 1 \tag{I53}$$

$$= \frac{2+b}{1+b} - 1 = \frac{1}{1+b} \,. \tag{I54}$$

The expectation value of the linear cross entropy is 1 when allowing repeated bitstrings if sampling from a Haar random quantum state. Therefore, the perturbation to the linear cross entropy when requiring unique bitstrings can be ignored when $b \ll 1$ or, equivalently, $\beta \ll D$. Note that sampling with less fidelity results in a lower frequency of collisions.

c. Adversarial postselection of repetitions

Consider now the situation where an adversarial server is asked to return k unique bitstrings, but the server secretly oversamples many more bitstrings to take advantage of collisions. That is, the server can postselect bitstrings that appear at least twice, and therefore, in the ideal case of fidelity 1, have a higher expectation value for the simulated probability $DA_{\beta,2} \sim 3$, instead of the usual average simulated probability $\langle Dp \rangle = 2$. In this way, the server could pass the linear cross entropy test returning a smaller number of quantum generated bitstrings, that is, a sample with similar estimated fidelity but less quantum entropy.

Next we bound how many bitstrings can be oversampled with still a negligible effect in the estimated fidelity from linear cross entropy. In order to cover the case of an adversarial server with non-ideal fidelity $\phi < 1$, we consider an idealized model where errors are heralded. That is, we treat sampling sk bitstrings with fidelity ϕ as sampling $\beta = sq$ bitstrings with fidelity 1, where $q = \phi k$. The contribution to the linear cross entropy for bitstrings that appear $c = 2$ times is

$$\frac{D}{q} M_{\beta,2} A_{\beta,2} = \frac{D}{q} \frac{\beta^2}{D^2} 3(1 + O(\beta/D)) \tag{I55}$$

$$= 2 \frac{s^2 q}{D} 3(1 + O(\beta/D)) \,. \tag{I56}$$

This effect is negligible for $s \ll \sqrt{D/q}$.

3. Additional statistical tests

We explained in the main text the conditions under which XEB is an estimator of fidelity, which is the main

test for experimental RCS (see also SM A). Ref. [6] also introduced the idea of checking the consistency between log and linear XEB, and a Kolmogorov-Smirnov test for the simulated probabilities of the experimental bitstrings. We now introduce two additional statistical tests which might be useful in an adversarial setting such as client-certified randomness generation.

a. *Hamming distance filter*

In order to sample from the output distribution of a quantum circuit one can use an independent tensor contraction per output bitstring using frugal rejection sampling (see SM G and Refs. [19, 30]). This results in a simulation runtime that scales linearly in the number of bitstrings sampled. Ref. [27] introduced a method to compute amplitudes of a large number of uncorrelated bitstrings with a much lower overhead than linear.

An adversarial server using tensor network contractions could avoid the remaining overhead from Ref. [27] using less tensor network contractions to calculate the probabilities of many bitstring with small Hamming distance between them, although this does not perform RCS (see for instance Ref. [51]). We now give a Hamming distance filter test which detects this pseudo-sampling.

We can approximate a Porter-Thomas sampling as a uniform sampling of bitstrings for the purpose of analyzing the Hamming distance between unique sampled bitstrings. We denote the distance between bitstrings j and k as h_{jk}. For fixed j, the distribution of the Hamming distance to other bitstrings is binomial with n the number of qubits and $p = 1/2$. As an example we can consider $n = 70$ and Hamming distance 15. The probability of $h_{jk} \leq 15$ is

$$p_h = \frac{1}{2^n} \sum_{c=0}^{15} \binom{70}{c} = 8.26 \cdot 10^{-7} \,. \tag{I57}$$

The experimental readout measurement error has a bias which we can take into account. Let e_{01} bet the probability of measuring state 0 when the quantum state is 1 and e_{10} the probability of measuring state 1 when the quantum state is 0. This gives a bias $b = e_{01} - e_{10}$. The probability of sampling a 1 on a qubit is, on average, $p_b = (1 - b)/2$, while the probability of sampling a 0 is $1 - p_b$. The probability of obtaining a given Hamming distance between two bitstrings is therefore given by a binomial distribution with the slightly biased value of p_b, which is slightly higher than in the unbiased case.

For a given sample S with k bitstrings, we can eliminate sufficient bitstrings so that there are no pairs of bitstrings within Hamming distance less than some bound, such as 15. One way to do this is to process the bitstrings one by one in the sample S. For each bitstring, we eliminate all the other bitstrings at Hamming distance 15 or less. With this method, we keep more than half of the bitstrings if $k = 10^6$. Note that each random ordering of bitstrings results in a different sub-sample. Therefore, this is equivalent to implementing bootstrapping in the initial sample. That is, we can repeat this sub-sampling a large number of times calculating the XEB fidelity estimator each time. The average of the XEB of all the sub-samples corresponds, in the honest case, to the sample average. We can do this for larger Hamming distances also.

In conclusion, with some small computational cost we can prevent a potential attack using a tensor network algorithm to calculate probabilities of sets bitstrings with small Hamming distance between them.

b. *Statistical test of large probabilities*

The value of the XEB fidelity estimator is higher if, instead of sampling, an adversarial server outputs the bitstrings with the highest simulated probabilities. This might be detected already by tests introduced in Ref. [6], such comparing linear and log XEB, or the Kolmogorov-Smirnov test. Here we give another option, namely using a truncated XEB fidelity estimator which ignores the bitstrings with simulated probability beyond some threshold t.

Consider the XEB estimator based on the function (see SM A)

$$f_t(p_j) := D\, p_j\, \mathbb{1}_{p_j \leq t/D} \tag{I58}$$

where p_j is the simulated or ideal probability for bitstring j, $D = 2^n$ is the Hilbert space dimension, and $\mathbb{1}_{p_j \leq t}$ is an indicator function with value 1 if $p_j \leq t$ and value 0 in other case. As explained in SM A we can model the sampling probabilities of a quantum processor with fidelity F as

$$p_j^F = F p_j + \frac{1 - F}{D} \,. \tag{I59}$$

The expectation value of the sampling with function (I58) is

$$\sum_j p_j^F\, D\, p_j \mathbb{1}_{p_j \leq t/D} \,. \tag{I60}$$

Assuming that the simulated probabilities are distributed according to the Porter-Thomas or exponential distribution we have

$$\begin{aligned} \text{tXEB} &= \sum_j p_j^F\, D\, p_j \mathbb{1}_{p_j \leq t/D} \\ &= D\, \langle\langle p_j^F\, D\, p_j \mathbb{1}_{p_j \leq t/D} \rangle\rangle & \text{(I61)} \\ &= D^2 \int_0^t (Fx + 1 - F) x e^{-x} dx & \text{(I62)} \\ &= F + 1 - e^{-t} \left(1 + t + (1 + t + t^2)F\right) \,. & \text{(I63)} \end{aligned}$$

As with any other XEB fidelity estimator, we can estimate the value tXEB sampling bitstrings from an experimental implementation. This gives the tXEB fidelity

estimator

$$F = \frac{\text{tXEB} - 1 + e^{-t}(1+t)}{1 - e^{-t}(1+t+t^2)} \,. \tag{I64}$$

Note that we are interested in the case $t \gtrsim 2$, which makes the denominator positive.

In order to calculate the variance of this estimators, we need the expectation value of the square of tXEB. This is

$$\sum_j p_j^F \left(D\, p_j \mathbb{1}_{p_j \le t/D}\right)^2 = D^2 \int_0^t (Fx + 1 - F) x^2 e^{-x} dx \,. \tag{I65}$$

This integral can be calculated analytically to obtain an expression for the variance of the corresponding estimator of fidelity. For simplicity, we only give the variance in the limit $F \to 0$, which is

$$\text{VAR(tXEB)} \simeq \frac{1 - e^{-t}t^2 - e^{-2t}(1+2t+t^2)}{\left(1 - e^{-t}(1+t+t^2)\right)^2} \,. \tag{I66}$$

Table I shows the variance for different values of the truncation parameter. We see that in an experimental sample we can ignore all bitstrings with ideal probabilities $\geq 4/D$ without affecting much the variance of the corresponding tXEB estimator. This gives another statistical test sensitive to a potential adversary which postselects bitstrings with unusually large ideal probabilities.

VAR(F)	t
1.00557	10.
1.06288	7.
1.32601	5.
1.84472	4.
4.11632	3.
10.144	2.5
105.982	2.

TABLE I. Variance of the truncated XEB estimate of F against the truncation parameter t.

4. Randomness extractor

Randomness extractors are functions that convert bits from a weak source of randomness into near-uniform random bits [52]. In our protocol we apply a randomness extractor to the output of a quantum computer, which contains intrinsic randomness but is not uniformly distributed. In this section, we describe the randomness extractor we implemented and present some benchmark results of its running time.

For general sources of randomness, randomness extraction is only possible if the extractor is also given a small uniformly random input seed as a catalyst.

A weak random source has a distribution over $\{0,1\}^n$ which has some entropy. The most conservative estimate of the unpredictability of the outcomes is given by the min-entropy or equivalently the $\infty-$Rényi entropy:

Definition 1. *Let X be a probability distribution on the hyper-cube $\{0,1\}^n$, and let p_x be the probability of the string $x \in \{0,1\}^n$. The minimum entropy of X is*

$$\min_x(-\log_2 p_x) = -\max_x \log_2 p_x = -\log_2 \max_x p_x$$

For an $n-$bit distribution X with min-entropy k, we say that X is an (n,k) distribution.

We now formally define an extractor function. Let $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be the function that takes as input samples from an (n,k) distribution X and a uniformly random $d-$bit string seed, and outputs an $m-$bit string that is ε-close to uniform. We say that the extractor is (k,ε) if the output is ε close to uniform random, where ε is the statistical distance. In extracting randomness from random variables from the knowledge of just a lower-bound on the min-entropy, a key concept is $k-$source.

Definition 2. *A random variable X is called a $k-$source if its min-entropy is at least k. That is, $Pr[X = x] \geq 2^{-k}$.*

a. *Trevisan's extractor and HMAC*

We implemented a randomness extractor based on Trevisan's construction [49]. Since this extractor is somewhat slow, we describe in App. I 4 c an alternative construction using the cryptographic primitive hash-based message authentication code (HMAC) that is more efficient, though it is a heuristic, not a theoretically proven extractor like Trevisan's.

We implemented Trevisan's extractor following primarily the construction of Raz, Reingold, and Vadhan [53] with some optimizations from [54]. The initial extractor was implemented entirely in Python, and profiling was used to identify bottlenecks, which were then rewritten in C++. In our case, over 99% of the running time was spent evaluating polynomials in a subroutine that computed Reed-Solomon codes. This code was migrated to a C++ library using NTL [55].

For a fixed ε, the extractor uses $\text{O}(\log^2 n)$ additional random bits. The theoretical optimal seed size for any seeded extractor, is $\log(n-k) + 2\log(2/\varepsilon) + \text{O}(1)$. For a fixed seed size, min-entropy, and ε, the extractor takes time linear in the size n of the input. However, our inputs are several orders of magnitude larger than those considered by previous papers and previous benchmarked implementations of Trevisan's extractor, such as [54] and [56].

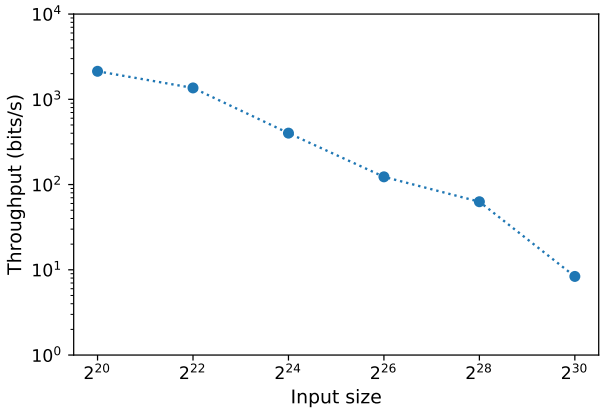


FIG. 23. Throughput of the Trevisan randomness extractor for various input sizes, when the output length is 4096 bits. We used 64 threads. Each data point is the average of the values obtained from 10 runs, and there are error bars of one standard deviation (too small to see).

b. *Benchmark results*

We tested the performance of our implementation of Trevisan's randomness extractor on various input sizes ranging from 2^{20} bits to 2^{30} bits. We used a workstation with an Intel Xeon Gold 6154 3.00GHz CPU which has 18 cores, each with 4 threads. We used 64 threads for our benchmarking.

Part of the running time of our extractor is spent in passing the input data from Python to C++. This conversion occurred at a rate of about 44 Mbit/s. Once this conversion has taken place, the extractor produces output bits at a constant rate, which we term the *throughput*, when the total length of the output is fixed. In Figure 23, we plot the throughput for the various input sizes when the output length is fixed to 4096 bits. At the input size of 2^{30}, the throughput was 8.4 bits/s. As an example, at the input size of 2^{30}, the Python to C++ conversion took about 24 seconds, while the rest of the extraction took about 490 seconds to produce the 4096 bits of output. While this throughput is slow compared to, say, the computation of a hash function like SHA-512, it is sufficient for many purposes. For example, a high-security cryptographic key requiring 256 bits of entropy may be used for days or weeks before needing to be refreshed.

c. *A faster randomness extractor using HMAC*

Our implementation of Trevisan's randomness extractor suffers from the disadvantage of being quite slow. In practice, theoretically proven randomness extractors are rarely used, with common efficient heuristic cryptographic primitives such as HMAC often used instead [57, 58]. In this appendix, we explain how one can use an HMAC to construct a heuristic randomness extractor that works in our setting. Besides being a heuristic, our construction suffers from the disadvantage of requiring a rather large seed size (linear in the size of the output). Nevertheless, it may be of more practical use in some situations than the Trevisan extractor.

The main obstacle to overcome in using an HMAC for randomness extraction is that the output length is limited. For example, a SHA512-based HMAC will only output 512 bits, even when the input has many more bits of min-entropy. Here, we show, using Lemma 6.38 in [52], that one can extend the output length of a randomness extractor. The lemma says the following:

Lemma 3. *(Lemma 6.38 in [52]): Suppose* $Ext_1\colon \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{m_1}$ *is a* (k_1,ε_1) *extractor and* $Ext_2\colon \{0,1\}^n \times \{0,1\}^{d_2} \to \{0,1\}^{m_2}$ *is a* (k_2,ε_2) *extractor for* $k_2 = k_1 - m_1 - \log(1/\varepsilon_3)$*. Then* $Ext'\colon \{0,1\}^n \times \{0,1\}^{d_1+d_2} \to \{0,1\}^{m_1+m_2}$ *defined by* $Ext'(x,(y_1,y_2)) = (Ext_1(x,y_1), Ext_2(x,y_2))$ *is a* $(k_1, \varepsilon_1+\varepsilon_2+\varepsilon_3)$ *extractor.*

Changing notation, letting $\mathrm{Ext}(X,S) := \mathrm{Ext}_1 = \mathrm{Ext}_2$, $m := m_1 = m_2$, and $\varepsilon := \varepsilon_1 = \varepsilon_2$; and renaming the independent variable ε_3 as ε_1 we can restate this lemma as: if $\mathrm{Ext}(X,S) = Y$ is a (k,ε)-extractor with an m-bit output, then $(\mathrm{Ext}(X,S_1), \mathrm{Ext}(X,S_2))$ is a $(k+m+\log(1/\varepsilon_1), 2\varepsilon+\varepsilon_1)$-extractor with a $2m$-bit output.

Let us define U_d to be a uniform random seed of size d bits. As explained in [52], $k_2 = k_1 - m_1 - \log(1/\varepsilon_3)$ in Lemma 3 arises because if one conditions a k_1-source on the output of $\mathrm{Ext}_1(X,U_{d_1})$, then the source still has a conditional min-entropy of at least $k_1 - m_1 - \log(1/\varepsilon_3) = k_2$ except with probability ε_3. Therefore, $\mathrm{Ext}_2(X,U_{d_2})$ can extract an additional m_2 almost-uniform bits. We can also ensure that $\mathrm{Ext}_2(X,U_{d_2})$ can extract an additional m_2 almost-uniform bits by instead requiring X to be a $(k_2+m_1+\log(1/\varepsilon_3))$-source.

This analysis can be recursively applied.

$$\begin{aligned}&\mathrm{Ext}''(X,(S_1,S_2,S_3,S_4))\\ &\quad\equiv (\mathrm{Ext}(X,S_1), \mathrm{Ext}(X,S_2), \mathrm{Ext}(X,S_3), \mathrm{Ext}(X,S_4))\end{aligned} \tag{I67}$$

can be considered to be the combination of two $(k+m+\log(1/\varepsilon_1), 2\varepsilon+\varepsilon_1)$-extractors:

- the first $(k+m+\log(1/\varepsilon_1), 2\varepsilon+\varepsilon_1)$-extractor is $(\mathrm{Ext}(X,S_1), \mathrm{Ext}(X,S_2))$ and
- the second $(k+m+\log(1/\varepsilon_1), 2\varepsilon+\varepsilon_1)$-extractor is $(\mathrm{Ext}(X,S_3), \mathrm{Ext}(X,S_4))$

Thus, selecting a new ε_2, $\mathrm{Ext}''(X,(S_1,S_2,S_3,S_4))$ is a $(k'+m'+\log(1/\varepsilon_2), 2\varepsilon'+\varepsilon_2)$-extractor, where

- $k' = k+m+\log(1/\varepsilon_1)$,
- $m' = 2m$, and
- $\varepsilon' = 2\varepsilon+\varepsilon_1$.

So we get a $\left(k+3m+\log\left(\frac{1}{\varepsilon_1}\right)+\log\left(\frac{1}{\varepsilon_2}\right), 4\varepsilon+2\varepsilon_1+\varepsilon_2\right)$-extractor.

In general, to output $2^t m$ bits, we can construct an extractor

$$\begin{aligned}&\mathrm{Ext}^t(X,(S_1,S_2,...,S_{2^t}))\\&\quad\equiv(\mathrm{Ext}(X,S_1),\mathrm{Ext}(X,S_2),...,\mathrm{Ext}(X,S_{2^t})).\end{aligned}\tag{I68}$$

This would be lead to the following extractor

$$\left(k+(2^t-1)m+\sum_{i=1}^{t}\log(1/\varepsilon_i), 2^t\varepsilon+\sum_{i=1}^{t}2^{t-i}\varepsilon_i\right)$$

This analysis demonstrates that given sufficient min-entropy in the input X, we can repeatedly apply the same randomness extractor with a fresh seed to extract the desired number of output bits that are statistically close to uniform.

Google Quantum AI and Collaborators

A. Morvan[1,‡], B. Villalonga[1,‡], X. Mi[1,‡], S. Mandrà[1,2,3,‡], A. Bengtsson[1], P. V. Klimov[1], Z. Chen[1], S. Hong[1], C. Erickson[1], I. K. Drozdov[1,4], J. Chau[1], G. Laun[1], R. Movassagh[1], A. Asfaw[1], L. T.A.N. Brandão[5], R. Peralta[5], D. Abanin[1], R. Acharya[1], R. Allen[1], T. I. Andersen[1], K. Anderson[1], M. Ansmann[1], F. Arute[1], K. Arya[1], J. Atalaya[1], J. C. Bardin[1,6], A. Bilmes[1], G. Bortoli[1], A. Bourassa[1], J. Bovaird[1], L. Brill[1], M. Broughton[1], B. B. Buckley[1], D. A. Buell[1], T. Burger[1], B. Burkett[1], N. Bushnell[1], J. Campero[1], H.-S. Chang[1], B. Chiaro[1], D. Chik[1], C. Chou[1], J. Cogan[1], R. Collins[1], P. Conner[1], W. Courtney[1], A. L. Crook[1], B. Curtin[1], D. M. Debroy[1], A. Del Toro Barba[1], S. Demura[1], A. Di Paolo[1], A. Dunsworth[1], L. Faoro[1], E. Farhi[1], R. Fatemi[1], V. S. Ferreira[1], L. Flores Burgos[1], E. Forati[1], A. G. Fowler[1], B. Foxen[1], G. Garcia[1], É. Genois[1] W. Giang[1], C. Gidney[1], D. Gilboa[1], M. Giustina[1], R. Gosula[1], A. Grajales Dau[1], J. A. Gross[1], S. Habegger[1], M. C. Hamilton[1,7], M. Hansen[1], M. P. Harrigan[1], S. D. Harrington[1], P. Heu[1], M. R. Hoffmann[1], T. Huang[1], A. Huff[1], W. J. Huggins[1], L. B. Ioffe[1], S. V. Isakov[1], J. Iveland[1], E. Jeffrey[1], Z. Jiang[1], C. Jones[1], P. Juhas[1], D. Kafri[1], T. Khattar[1], M. Khezri[1], M. Kieferová[1,8], S. Kim[1], A. Kitaev[1], A. R. Klots[1], A. N. Korotkov[1,9], F. Kostritsa[1], J. M. Kreikebaum[1], D. Landhuis[1], P. Laptev[1], K.-M. Lau[1], L. Laws[1], J. Lee[1,10], K. W. Lee[1], Y. D. Lensky[1], B. J. Lester[1], A. T. Lill[1], W. Liu[1], W. P. Livingston[1], A. Locharla[1], F. D. Malone[1], O. Martin[1], S. Martin[1], J. R. McClean[1], M. McEwen[1], K. C. Miao[1], A. Mieszala[1], S. Montazeri[1], W. Mruczkiewicz[1], O. Naaman[1], M. Neeley[1], C. Neill[1], A. Nersisyan[1], M. Newman[1], J. H. Ng[1], A. Nguyen[1], M. Nguyen[1], M. Yuezhen Niu[1], T. E. O'Brien[1], S. Omonije[1], A. Opremcak[1], A. Petukhov[1], R. Potter[1], L. P. Pryadko[11], C. Quintana[1], D. M. Rhodes[1], C. Rocque[1], E. Rosenberg[1], N. C. Rubin[1], N. Saei[1], D. Sank[1], K. Sankaragomathi[1], K. J. Satzinger[1], H. F. Schurkus[1], C. Schuster[1], M. J. Shearn[1], A. Shorter[1], N. Shutty[1], V. Shvarts[1], V. Sivak[1], J. Skruzny[1], W. C. Smith[1], R. D. Somma[1], G. Sterling[1], D. Strain[1], M. Szalay[1], D. Thor[1], A. Torres[1], G. Vidal[1], C. Vollgraff Heidweiller[1], T. White[1], B. W. K. Woo[1], C. Xing[1], Z. J. Yao[1], P. Yeh[1], J. Yoo[1], G. Young[1], A. Zalcman[1], Y. Zhang[1], N. Zhu[1], N. Zobrist[1], E. G. Rieffel[2], R. Biswas[2], R. Babbush[1], D. Bacon[1], J. Hilton[1], E. Lucero[1], H. Neven[1], A. Megrant[1], J. Kelly[1], P. Roushan[1], I. Aleiner[1], V. Smelyanskiy[1], K. Kechedzhi[1,§], Y. Chen[1,§], S. Boixo[1,§],

[1] Google Research
[2] Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, California 94035, USA
[3] KBR, 601 Jefferson St., Houston, TX 77002, USA
[4] Department of Physics, University of Connecticut, Storrs, CT
[5] National Institute of Standards and Technology (NIST), USA
[6] Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA
[7] Department of Electrical and Computer Engineering, Auburn University, Auburn, AL
[8] QSI, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia
[9] Department of Electrical and Computer Engineering, University of California, Riverside, CA
[10] Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA, USA
[11] Department of Physics and Astronomy, University of California, Riverside, CA
‡ These authors contributed equally to this work.

[1] W. K. Wootters, Random quantum states, Foundations of Physics **20**, 1365 (1990).
[2] C. M. Caves, Measures and volumes for spheres, the probability simplex, projective hilbert space, and density operators, Unpublished (2001).
[3] D. Petz and J. Réffy, On asymptotics of large haar distributed unitary matrices, Periodica Mathematica Hungarica **49**, 103 (2004).
[4] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Characterizing quantum supremacy in near-term devices, Nature Physics **14**, 595 (2018).
[5] K. Zyczkowski and H.-J. Sommers, Induced measures in the space of mixed quantum states, Journal of Physics A: Mathematical and General **34**, 7111 (2001).
[6] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro,

R. Collins, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[7] R. Barends, C. M. Quintana, A. G. Petukhov, Y. Chen, D. Kafri, K. Kechedzhi, R. Collins, O. Naaman, S. Boixo, F. Arute, K. Arya, D. Buell, B. Burkett, Z. Chen, B. Chiaro, A. Dunsworth, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, T. Huang, E. Jeffrey, J. Kelly, P. V. Klimov, F. Kostritsa, D. Landhuis, E. Lucero, M. McEwen, A. Megrant, X. Mi, J. Mutus, M. Neeley, C. Neill, E. Ostby, P. Roushan, D. Sank, K. J. Satzinger, A. Vainsencher, T. White, J. Yao, P. Yeh, A. Zalcman, H. Neven, V. N. Smelyanskiy, and J. M. Martinis, Diabatic gates for frequency-tunable superconducting qubits, Physical Review Letters **123**, 210501 (2019).

[8] C. Neill, T. McCourt, X. Mi, Z. Jiang, M. Y. Niu, W. Mruczkiewicz, I. Aleiner, F. Arute, K. Arya, J. Atalaya, R. Babbush, J. C. Bardin, R. Barends, A. Bengtsson, A. Bourassa, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, J. Campero, Z. Chen, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. R. Derk, A. Dunsworth, D. Eppens, C. Erickson, E. Farhi, A. G. Fowler, B. Foxen, C. Gidney, M. Giustina, J. A. Gross, M. P. Harrigan, S. D. Harrington, J. Hilton, A. Ho, S. Hong, T. Huang, W. J. Huggins, S. V. Isakov, M. Jacob-Mitos, E. Jeffrey, C. Jones, D. Kafri, K. Kechedzhi, J. Kelly, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, E. Lucero, O. Martin, J. R. McClean, M. McEwen, A. Megrant, K. C. Miao, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, M. Newman, T. E. O'Brien, A. Opremcak, E. Ostby, B. Pató, A. Petukhov, C. Quintana, N. Redd, N. C. Rubin, D. Sank, K. J. Satzinger, V. Shvarts, D. Strain, M. Szalay, M. D. Trevithick, B. Villalonga, T. C. White, Z. Yao, P. Yeh, A. Zalcman, H. Neven, S. Boixo, L. B. Ioffe, P. Roushan, Y. Chen, and V. Smelyanskiy, Accurately computing the electronic properties of a quantum ring, Nature **594**, 508 (2021).

[9] P. V. Klimov, J. Kelly, J. M. Martinis, and H. Neven, The snake optimizer for learning quantum processor control parameters (2020), arXiv:2006.04594.

[10] I. L. Aleiner, L. Faoro, and L. B. Ioffe, Microscopic model of quantum butterfly effect: out-of-time-order correlators and traveling combustion waves, Annals of Physics **375**, 378 (2016).

[11] A. Nahum, S. Vijay, and J. Haah, Operator spreading in random unitary circuits, Phys. Rev. X **8**, 021014 (2018).

[12] C. W. von Keyserlingk, T. Rakovszky, F. Pollmann, and S. L. Sondhi, Operator hydrodynamics, otocs, and entanglement growth in systems without conservation laws, Phys. Rev. X **8**, 021013 (2018).

[13] X. Mi, P. Roushan, C. Quintana, S. Mandra, J. Marshall, C. Neill, F. Arute, K. Arya, J. Atalaya, R. Babbush, et al., Information scrambling in quantum circuits, Science **374**, 1479 (2021).

[14] X. Gao, M. Kalinowski, C.-N. Chou, M. D. Lukin, B. Barak, and S. Choi, Limitations of linear cross-entropy as a measure for quantum advantage, PRX Quantum **5**, 010334 (2024).

[15] A. M. Dalzell, N. Hunter-Jones, and F. G. S. L. Brandão, Random quantum circuits anticoncentrate in log depth, PRX Quantum **3**, 010333 (2022).

[16] B. Barak, C.-N. Chou, and X. Gao, Spoofing linear cross-entropy benchmarking in shallow quantum circuits, arXiv:2005.02421 (2020).

[17] D. Aharonov, X. Gao, Z. Landau, Y. Liu, and U. Vazirani, A polynomial-time classical algorithm for noisy random circuit sampling, in Proceedings of the 55th Annual ACM Symposium on Theory of Com (2023) pp. 945–957.

[18] P. de Gennes and J. Prost, The Physics of Liquid Crystals, International Series of Monographs on Physics (Clarendon Press, 1993).

[19] I. L. Markov, A. Fatima, S. V. Isakov, and S. Boixo, Quantum supremacy is both closer and farther than it appears, arXiv:1807.10749 (2018).

[20] B. Bertini, P. Kos, and T. Prosen, Operator entanglement in local quantum circuits i: Chaotic dual-unitary circuits, SciPost Physics **8**, 067 (2020).

[21] I. L. Markov and Y. Shi, Simulating quantum computation by contracting tensor networks, SIAM Journal on Computing **38**, 963 (2008).

[22] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, and H. Neven, Simulation of low-depth quantum circuits as complex undirected graphical models, arXiv:1712.05384 (2017).

[23] J. Gray and S. Kourtis, Hyper-optimized tensor network contraction, Quantum **5**, 410 (2021).

[24] C. Huang, F. Zhang, M. Newman, J. Cai, X. Gao, Z. Tian, J. Wu, H. Xu, H. Yu, B. Yuan, M. Szegedy, Y. Shi, and J. Chen, Classical simulation of quantum supremacy circuits, arXiv:2005.06787 (2020).

[25] G. Kalachev, P. Panteleev, and M.-H. Yung, Multi-tensor contraction for xeb verification of quantum circuits, arXiv:2108.05665 (2021).

[26] G. Kalachev, P. Panteleev, P. Zhou, and M.-H. Yung, Classical sampling of random quantum circuits with bounded fidelity, arXiv:2112.15083 (2021).

[27] F. Pan, K. Chen, and P. Zhang, Solving the sampling problem of the sycamore quantum circuits, Physical Review Letters **129**, 090502 (2022).

[28] Y. Liu, Y. Chen, C. Guo, J. Song, X. Shi, L. Gan, W. Wu, W. Wu, H. Fu, X. Liu, D. Chen, G. Yang, and J. Gao, Validating quantum-supremacy experiments with exact and fast tensor network contraction, arXiv:2212.04749 (2022).

[29] J. Chen, F. Zhang, C. Huang, M. Newman, and Y. Shi, Classical simulation of intermediate-size quantum circuits, arXiv:1805.01450 (2018).

[30] B. Villalonga, S. Boixo, B. Nelson, C. Henze, E. Rieffel, R. Biswas, and S. Mandrà, A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware, npj Quantum Information **5**, 86 (2019).

[31] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen,

X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, S. Li, Y. Li, F. Liang, C. Lin, J. Lin, H. Qian, D. Qiao, H. Rong, H. Su, L. Sun, L. Wang, S. Wang, D. Wu, Y. Xu, K. Yan, W. Yang, Y. Yang, Y. Ye, J. Yin, C. Ying, J. Yu, C. Zha, C. Zhang, H. Zhang, K. Zhang, Y. Zhang, H. Zhao, Y. Zhao, L. Zhou, Q. Zhu, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, Strong quantum computational advantage using a superconducting quantum processor, Physical Review Letters **127**, 180501 (2021).

[32] Q. Zhu, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, S. Li, Y. Li, F. Liang, C. Lin, J. Lin, H. Qian, D. Qiao, H. Rong, H. Su, L. Sun, L. Wang, S. Wang, D. Wu, Y. Wu, Y. Xu, K. Yan, W. Yang, Y. Yang, Y. Ye, J. Yin, C. Ying, J. Yu, C. Zha, C. Zhang, H. Zhang, K. Zhang, Y. Zhang, H. Zhao, Y. Zhao, L. Zhou, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, Quantum computational advantage via 60-qubit 24-cycle random circuit sampling, Science Bulletin **67**, 240 (2022).

[33] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers?, Physical Review X **10**, 041038 (2020).

[34] T. Ayral, 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).

[35] Ref. [34] claims that using MPS methods the linear XEB scales as the square root of the fidelity, which contradicts our findings. This is because a) at low number of cycles they compute XEB before the anti-concentration point and b) at high number of cycles they plot the absolute value of the linear XEB (instead of the linear XEB itself).

[36] R. Oliveira, O. Dahlsten, and M. Plenio, Generic entanglement can be generated efficiently, Physical Review Letters **98**, 130502 (2007).

[37] V. A. Marčenko and L. A. Pastur, Distribution of eigenvalues for some sets of random matrices, Mathematics of the USSR-Sbornik **1**, 457 (1967).

[38] R. Movassagh and A. Edelman, Isotropic entanglement, arXiv:1012.5039 (2010).

[39] We study the circuit ensemble consisting of random choices of one-qubit gates $Z^pX^{1/2}Z^{-p}$ with $p \in \{-1, -1/4, -1/2, \ldots, 3/4\}$ and the two-qubit gate iSWAP^{-1}. We obtain the same ensemble if we use iSWAP instead of iSWAP^{-1}. This follows from $\mathrm{iSWAP}^{-1} = \mathrm{iSWAP} \cdot (Z \otimes Z) = (Z \otimes Z) \cdot \mathrm{iSWAP}$ and the fact that the set $\{ZZ^pX^{1/2}Z^{-p}\}$ is the same as $\{Z^pX^{1/2}Z^{-p}Z\}$. Therefore we can move Z gates between layers as we transform iSWAP^{-1}'s to iSWAP's.

[40] K. M. Audenaert and M. B. Plenio, Entanglement on mixed stabilizer states: normal forms and reduction procedures, New Journal of Physics **7**, 170 (2005).

[41] S. Aaronson, Certified randomness from quantum supremacy, Talk at CRYPTO 2018 (2018).

[42] R. Bassirian, A. Bouland, B. Fefferman, S. Gunn, and A. Tal, On certified randomness from quantum advantage experiments, arXiv:2111.14846 (2021).

[43] S. Aaronson and S.-H. Hung, Certified randomness from quantum supremacy, in Proceedings of the 55th Annual ACM Symposium on Theory of Com (2023) pp. 933–944.

[44] A. Morningstar, M. Hauru, J. Beall, M. Ganahl, A. G. Lewis, V. Khemani, and G. Vidal, Simulation of quantum many-body dynamics with tensor processing units: Floquet prethermalization, PRX Quantum **3**, 020331 (2022).

[45] Z. Brakerski, P. Christiano, U. Mahadev, U. Vazirani, and T. Vidick, A cryptographic test of quantumness and certifiable randomness from a single quantum device, in 2018 IEEE 59th Annual Symposium on Foundations of Computer Sc (IEEE, 2018) pp. 320–331.

[46] U. Mahadev, U. Vazirani, and T. Vidick, Efficient certifiable randomness from a single quantum device, arXiv:2204.11353 (2022).

[47] M.-H. Yung and B. Cheng, Anti-forging quantum data: Cryptographic verification of quantum computational power, arXiv:2005.01510 (2020).

[48] The parameter b in the "Linear Cross-Entropy Benchmarking" definition of Problem 1 in Ref. [43] corresponds to our XEB + 1. Note that this parameter b is assumed to be constant as the number of qubits n scales.

[49] L. Trevisan, Extractors and pseudorandom generators, Journal of the ACM **48**, 860 (2001).

[50] L. Brandão and R. Peralta, Notes on interrogating random quantum circuits, National Institute of Standards and Technology (2020), doi: 10.13140/RG.2.2.24562.94400.

[51] F. Pan and P. Zhang, Simulating the sycamore quantum supremacy circuits, arXiv:2103.03074 (2021).

[52] S. P. Vadhan, Pseudorandomness, Foundations and Trends in Theoretical Computer Science **7**, 1 (2012).

[53] R. Raz, O. Reingold, and S. Vadhan, Extracting all the randomness and reducing the error in Trevisan's extractors, Journal of Computer and System Sciences **65**, 97 (2002).

[54] W. Mauerer, C. Portmann, and V. B. Scholz, A modular framework for randomness extraction based on trevisan's construction, arXiv:1212.0520 (2012).

[55] V. Shoup, NTL: A library for doing number theory, https://www.shoup.net/ntl/download.html (2020).

[56] X. Ma, F. Xu, H. Xu, X. Tan, B. Qi, and H.-K. Lo, Postprocessing for quantum random-number generators: Entropy evaluation and randomness extraction, Physical Review A **87**, 062327 (2013).

[57] National Institute of Standards and Technology, FIPS PUB 198-1: The Keyed-Hash Message Authentication Code (H (National Institute for Standards and Technology, 2008).

[58] Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin, Randomness extraction and key derivation using the cbc, cascade and hmac modes, in Advances in Cryptology – CRYPTO 2004, edited by M. Franklin (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004) pp. 494–510.