Characterizing Quantum Supremacy in Near-Term Devices

Sergio Boixo,[1] Sergei V. Isakov,[2] Vadim N. Smelyanskiy,[1] Ryan Babbush,[1]
Nan Ding,[1] Zhang Jiang,[3,4] John M. Martinis,[5,6] and Hartmut Neven[1]

[1] *Google Inc., Venice, CA 90291, USA*
[2] *Google Inc., 8002 Zurich, Switzerland*
[3] *QuAIL, NASA Ames Research Center, Moffett Field, CA 94035, USA*
[4] *SGT Inc., 7701 Greenbelt Rd., Suite 400, Greenbelt, MD 20770*
[5] *Google Inc., Santa Barbara, CA 93117, USA*
[6] *Department of Physics, University of California, Santa Barbara, CA 93106, USA*

(Dated: August 4, 2016)

A critical question for the field of quantum computing in the near future is whether quantum devices without error correction can perform a well-defined computational task beyond the capabilities of state-of-the-art classical computers, achieving so-called quantum supremacy. We study the task of sampling from the output distributions of (pseudo-)random quantum circuits, a natural task for benchmarking quantum computers. Crucially, sampling this distribution classically requires a direct numerical simulation of the circuit, with computational cost exponential in the number of qubits. This requirement is typical of chaotic systems. We extend previous results in computational complexity to argue more formally that this sampling task must take exponential time in a classical computer. We study the convergence to the chaotic regime using extensive supercomputer simulations, modeling circuits with up to 42 qubits - the largest quantum circuits simulated to date for a computational task that approaches quantum supremacy. We argue that while chaotic states are extremely sensitive to errors, quantum supremacy can be achieved in the near-term with approximately fifty superconducting qubits. We introduce cross entropy as a useful benchmark of quantum circuits which approximates the circuit fidelity. We show that the cross entropy can be efficiently measured when circuit simulations are available. Beyond the classically tractable regime, the cross entropy can be extrapolated and compared with theoretical estimates of circuit fidelity to define a practical quantum supremacy test.

I. INTRODUCTION

Despite a century of research, there is no known method for efficiently simulating arbitrary quantum dynamics using classical computation. In practice, we are unable to directly simulate even modest depth quantum circuits acting on approximately fifty qubits. This strongly suggests that the controlled evolution of ideal quantum systems offers computational resources more powerful than classical computers [1, 2]. In this paper we build on existing results in quantum chaos [3–17] and computational complexity theory [18–28] to propose an experiment for characterizing “quantum supremacy” [29] in the presence of errors. We study the computational task of sampling from the output distribution of random quantum circuits composed from a universal gate set, a natural task for benchmarking quantum computers. We propose the cross entropy difference as a measure of correspondence between experimentally obtained samples and the output distribution of the ideal circuit. Finally, we discuss a robust set of conditions which should be met in order to be sufficiently confident that an experimental demonstration has actually achieved quantum supremacy.

In this paper we show how to estimate the cross entropy between an experimental implementation of a random quantum circuit and the ideal output distribution simulated by a supercomputer. We study numerically the convergence of the output distribution to the Porter-Thomas distribution, characteristic of quantum chaos. We find a good convergence for the first ten moments and the entropy at depth 25 with circuits of up to 7×6 qubits in a 2D lattice. Using chaos theory, the properties of the Porter-Thomas distribution, and numerical simulations, we argue that the cross entropy is closely related to the circuit fidelity. State-of-the-art supercomputers cannot simulate universal random circuits with depth 25 in a 2D lattice of approximately 7×7 qubits with any known algorithm and significant fidelity.

Time accurate simulations of classical dynamical systems with chaotic behavior are among the hardest numerical tasks. Examples include turbulence and population dynamics, essential for the study of meteorology, biology, finance, etc. In all these cases, a direct numerical simulation is required in order to get an accurate description of the system state after a finite time. A signature of chaotic systems is that small changes in the model specification lead to large divergences in system trajectories. This phenomenon is described by Lyapunov exponents and generally requires computational resources that grow exponentially in time.

In quantum chaotic dynamics this sensitivity manifests itself in the decrease of the overlap $|\langle\psi_t|\psi_t^\epsilon\rangle|^2$ of the quantum state $|\psi_t\rangle$ with the state $|\psi_t^\epsilon\rangle$ resulting from a small perturbation ϵ to the Hamiltonian that evolves $|\psi_t\rangle$ [4, 5, 8]. The overlap decreases exponentially in ϵ because chaotic evolutions give rise to delocalization of quantum states [6, 7]. Such states are closely related to ensembles of random unitary matrices studied in random

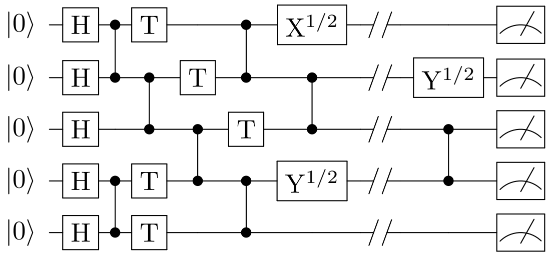

FIG. 1. Example of a random quantum circuit in a 1D array of qubits. Vertical lines correspond to controlled-phase (CZ) gates (see Sec. IV).

matrix theory [6, 30], they possess no symmetries, and are spread over Hilbert space. Therefore, as in the case of classical chaos, obtaining a description of $|\psi_t\rangle$ requires a high fidelity classical simulation. This challenge is compounded by the exponential growth of Hilbert space $N = 2^n$ with the qubit dimension n.

It follows that unless a classical algorithm uses resources that grow exponentially in n, its output would be almost statistically uncorrelated with the output distribution corresponding to general global measurements of the chaotic quantum state.[1] Indeed, it has been argued that classically solving related sampling problems requires computational resources with asymptotic exponential scaling [18–28]. Examples include Bosonsampling [22] and approximate simulation of commuting quantum computations [21, 27].

Random quantum circuits with gates sampled from a universal gate set are examples of quantum chaotic evolutions that naturally lend themselves to the quantum computational framework [7, 9–12, 14]. A circuit, corresponding to a unitary transformation U, is a sequence of d clock cycles of one- and two-qubit gates, with gates applied to different qubits in the same cycle, see Fig. 1. With realistic superconducting hardware constraints [31, 32], gates act in parallel on distinct sets of qubits restricted to a 1D or 2D lattice.

In this paper we study the computational task of sampling bit-strings from the distribution defined by the output state $|\psi\rangle$ of a (pseudo-)random quantum circuit U of size *polynomial* in n. We will compare the sampling output of U to a generic classical sampling algorithm that takes a specification of U as input and samples a bit-string with computational time cost also *polynomial* in n. We will show that a bit-string sampled from U is typically e times more likely than a bit-string sampled by the classical algorithm. A quantum sample S of m measurement outcomes $x \in \{0,1\}^n$ in a local qubit basis has probability $\Pi_{x\in S} |\langle x|\psi\rangle|^2$. Denote by $S_{\rm pcl}$ a sample of m bit-strings from the polynomial classical algorithm. We argued above from standard assumptions in chaos theory that in this case $S_{\rm pcl}$ is expected to be almost uncorrelated with the distribution defined by $|\psi\rangle$. The sample $S_{\rm pcl}$ is assigned a probability $\Pi_{x\in S_{\rm pcl}} |\langle x|\psi\rangle|^2$ by the distribution defined by $|\psi\rangle$. As we show in this paper, the ratio of these probabilities for a sufficiently large circuit in the typical case is, within logarithmic equivalence, $\Pi_{x\in S} |\langle x|\psi\rangle|^2 / \Pi_{x\in S_{\rm pcl}} |\langle x|\psi\rangle|^2 \sim e^m$ (see Eq. (9)). We will also show that for a typical sample $S_{\rm exp}$ produced by an experimental implementation of U this ratio is, within logarithmic equivalence,

$$\frac{\Pi_{x\in S_{\rm exp}} |\langle x|\psi\rangle|^2}{\Pi_{x\in S_{\rm pcl}} |\langle x|\psi\rangle|^2} \sim e^{m\, e^{-rg}} \gg 1\ , \tag{1}$$

where the parameter r provides an estimate of the effective per-gate error rate, and $g \propto nd$ is the total number of gates (see Eqs. (14) and (16)). Note the double exponential structure in Eq. (1) with two large parameters $m, g \gg 1$. Therefore, the ratio of probabilities in Eq. (1), an experimentally observable quantity, is enormously sensitive to the effective per-gate error rate r. The parameter r can serve as an extremely accurate characterization of the degree of correlation of $S_{\rm exp}$ with the distribution defined by U, and provides a novel tool for benchmarking complex multiqubit quantum circuits. We will argue that r can be estimated theoretically and compared with experiments to define a quantum supremacy test.

We now give the main outline of the paper. In Sec. II we obtain Eq. (1) from the cross entropy between the two distributions and we explain how it can be measured in an experiment. In Sec. III we explain theoretically and numerically why the cross entropy is closely related to the overall circuit fidelity. We also introduce an effective error model for the overall circuit, and compare it with numerical simulations of the circuit with digital errors. In Sec. IV we study numerically the convergence of the circuit output to the Porter-Thomas distribution, characteristic of quantum chaos. In Sec. V we use complexity theory to argue that this sampling problem is computational hard.

II. CHARACTERIZING QUANTUM SUPREMACY

A. Ideal circuit vs. polynomial classical algorithm

Consider a state $|\psi_d\rangle$ produced by a random quantum circuit. Due to delocalization, the real and imaginary parts of the amplitudes $\langle x_j|\psi_d\rangle$ in any local qubit basis $\{x_j\}_{j=1}^N$, $x_j \in \{0,1\}^n$ are approximately uniformly distributed in a $2N = 2^{n+1}$ dimensional sphere (Hilbert space) subject to the normalization constraint. This implies that their distribution is an unbiased Gaussian with

[1] A classical algorithm that uses time and space resources that grow exponentially in n can reconstruct all measurements of the chaotic quantum state exactly.

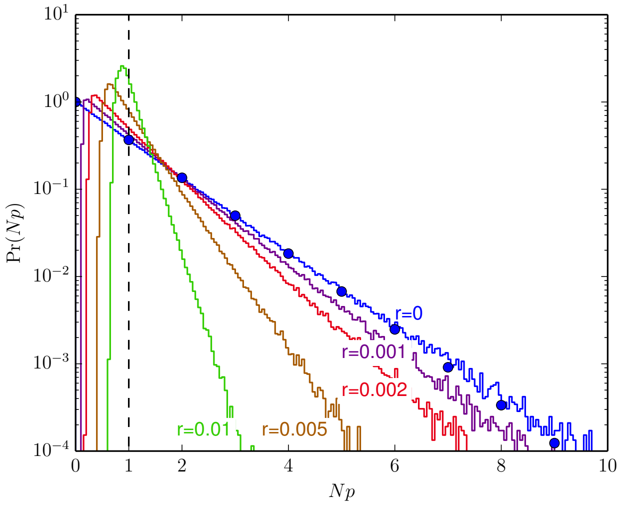

FIG. 2. Distribution function of rescaled probabilities Np to observe individual bit-strings as an output of a typical random circuit. Blue curve ($r = 0$) shows the distribution of $\{Np_U(x_j)\}$ obtained from numerical simulations of the ideal random circuit (see Sec. IV) . This distribution is very close to the Porter-Thomas form $P(Np) = e^{-Np}$ shown with blue dots. Curves with different colors show the distributions of probabilities obtained for different Pauli error rates r. The dashed line at $Np = 1$ corresponds to the uniform distribution $\delta(p - 1/N)$. These numerics are obtained from simulations of a planar circuit with 5×4 qubits and gate depth of 25 ($n = 20$ and $N = 2^{20}$).

variance $\propto 1/N$, up to finite moments [33]. This distribution is a signature of delocalization due to quantum correlations manifested as level repulsion in systems with stationary Hamiltonians. The distribution of measurement probabilities $p(x_j) = |\langle x_j|\psi_d\rangle|^2$ approaches the exponential form Ne^{-Np}, known as Porter-Thomas [3], see Fig. 2. The probability vectors thus obtained are uniformly distributed over the probability simplex (i.e., according to the symmetric Dirichlet distribution).

The circuit depth or time to approach the Porter-Thomas regime is expected to correspond to the ballistic spread of entanglement across Hilbert space in chaotic systems [16, 17]. This timescale grows as $n^{1/D}$ where D is the dimension of the qubit lattice. In particular, $D = 1$ for a linear array [34], $D = 2$ for a square lattice, and D goes to infinity for a fully connected architecture [12, 13, 15] (see Sec. IV).

The output probability $p(x_j)$ of each bit-string from a random quantum circuit is of order $1/N = 2^{-n}$, see Fig. 2. Therefore, each bit-string in a sample of size polynomial in n will be unique. In other words, the output of a random quantum circuit can not be distinguished from a uniform sampler over $\{x_j\}$ unless we pre-compute the specific output probabilities $p(x_j)$ [35–38].[2]

Nevertheless, the Porter-Thomas distribution Ne^{-Np} has substantial support on values $Np < 1$, see Fig. 2. This will allow us to clearly distinguish it from the uniform distribution over $\{x_j\}$, which has a form given by a delta function $\delta(p - 1/N)$, after computing $p(x_j)$ with a powerfull enough classical computer. Circuit specific global measurements can be sensitive to time-accurate simulations of chaotic quantum state evolutions.[3] Therefore, such observables will be extremely hard to simulate classically.

Let $|\psi\rangle = U|\psi_0\rangle$ be the output of a given random circuit U. Consider a sample $S = \{x_1, \ldots, x_m\}$ of bit-strings x_j obtained from m global measurements of every qubit in the computational basis $\{|x_j\rangle\}$ (or any other basis obtained from local operations). The joint probability of the set of outcomes S is $\Pr_U(S) = \prod_{x_j \in S} p_U(x_j)$ where $p_U(x) \equiv |\langle x|\psi\rangle|^2$. For a typical sample S, the central limit theorem implies that

$$\begin{aligned} \log \Pr_U(S) &= \sum_{x_j \in S} \log p_U(x_j) \\ &= -m\,\mathrm{H}(p_U) + O(m^{1/2})\ , \end{aligned} \tag{2}$$

where $\mathrm{H}(p_U) \equiv -\sum_{j=1}^{N} p_U(x_j) \log p_U(x_j)$ is the entropy of the output of U. Because $p_U(x)$ are approximately *i.i.d.* distributed according to the Porter-Thomas distribution, if follows that

$$\begin{aligned} \mathrm{H}(p_U) &= -\int_0^\infty pN^2 e^{-Np} \log p\, dp \\ &= \log N - 1 + \gamma\ , \end{aligned} \tag{3}$$

where $\gamma \approx 0.577$ is the Euler constant.

Let $A_{\rm pcl}(U)$ be a classical algorithm with computational time cost *polynomial* in n that takes a specification of the random circuit U as input and outputs a bit-string x with probability distribution $p_{\rm pcl}(x|U)$. Consider a typical sample $S_{\rm pcl} = \{x_1^{\rm pcl}, \ldots, x_m^{\rm pcl}\}$ obtained from $A_{\rm pcl}(U)$. We now focus on the probability $\Pr_U(S_{\rm pcl}) = \prod_{x_j^{\rm pcl} \in S_{\rm pcl}} p_U(x_j^{\rm pcl})$ that this sample $S_{\rm pcl}$ is observed from the output $|\psi\rangle$ of the circuit U. The central limit theorem implies that

$$\log \Pr_U(S_{\rm pcl}) = -m\,\mathrm{H}(p_{\rm pcl}, p_U) + O(m^{1/2})\ , \tag{4}$$

[2] In the case of Bosonsampling, generic observables sensitive to Boson statistics can be used to distinguish the output distribution from uniform [39, 40]. Nevertheless, it is also unlikely that a Bosonsampler can be distinguished from classically efficient simulations unless we use exponential resources [22, 39].

[3] Specifically, the ℓ_1 norm distance between the Porter-Thomas distribution and the uniform distribution over $\{x_j\}$ is $2/e$, independent of n. Therefore, information theoretically, a constant small number of measurements are sufficient to distinguish these distributions.

where

$$\mathrm{H}(p_{\mathrm{pcl}}, p_U) \equiv -\sum_{j=1}^{N} p_{\mathrm{pcl}}(x_j|U) \log p_U(x_j) \tag{5}$$

is the cross entropy between $p_{\mathrm{pcl}}(x|U)$ and $p_U(x)$. Note that if the cross entropy $\mathrm{H}(p_{\mathrm{pcl}}, p_U)$ is larger than the entropy $\mathrm{H}(p_U)$, this implies that $p_{\mathrm{pcl}}(x|U)$ is sampling bit-strings that have lower probability of being observed by the circuit U.

We are interested in the average quality of the classical algorithm. Therefore, we average the cross entropy over an ensemble $\{U\}$ of random circuits

$$\mathbb{E}_U\left[\mathrm{H}(p_{\mathrm{pcl}}, p_U)\right] = \mathbb{E}_U\left[\sum_{j=1}^{N} p_{\mathrm{pcl}}(x_j|U) \frac{1}{\log p_U(x_j)}\right] . \tag{6}$$

Based on aforementioned insights from quantum chaos, we assume that the output of a classical algorithm with polynomial cost is almost statistically uncorrelated with $p_U(x)$ (see also App. H). Thus, averaging over the ensemble $\{U\}$ can be done independently for the output of the polynomial classical algorithm $p_{\mathrm{pcl}}(x|U)$ and $\log p_U(x)$. The distribution of universal random quantum circuits converges to the uniform (Haar) measure with increasing depth [7, 12, 41]. For fixed x_j, the distribution of values $\{p_U(x_j)\}$ when unitaries are sampled from the Haar measure also has the Porter-Thomas form. Therefore, we assume that we use random circuits of sufficient depth such that

$$\begin{aligned} -\mathbb{E}_U\left[\log p_U(x_j)\right] &\approx -\int_0^\infty N e^{-Np} \log p \, dp \\ &= \log N + \gamma \; . \end{aligned} \tag{7}$$

Note that this equation is similar to Eq. (3), except that the integrand here is missing a factor of Np. Then using $\sum_{j=1}^{N} p_{\mathrm{pcl}}(x_j|U) = 1$ we get

$$\mathbb{E}_U\left[\mathrm{H}(p_{\mathrm{pcl}}, p_U)\right] = \log N + \gamma \; . \tag{8}$$

From Eqs. (3) and (8) we obtain

$$\mathbb{E}_U\left[\log \mathrm{Pr}_U(S) - \log \mathrm{Pr}_U(S_{\mathrm{pcl}})\right] \simeq m \; . \tag{9}$$

Equation (9) reveals the remarkable property that a typical sample S from a random circuit U represents a signature of that circuit. Note that the l.h.s. is the expectation value of the log of $\Pi_{x \in S} | \langle x|\psi\rangle |^2 / \Pi_{x^{\mathrm{pcl}} \in S_{\mathrm{pcl}}} | \langle x^{\mathrm{pcl}}|\psi\rangle |^2$. The numerator is dominated by measurement outcomes x that have high measurement probabilities $| \langle x|\psi\rangle |^2 > 1/N$. Conversely, the values of x^{pcl} in the denominator are essentially uncorrelated with the output distribution of U. Therefore, they are dominated by the support of the Porter-Thomas distribution with $p < 1/N$.

B. Cross entropy difference

We note that the result in Eq. (8) also corresponds to the cross entropy $\mathrm{H}_0 = \log N + \gamma$ of an algorithm which picks bit-strings uniformly at random, $p_0(x) = 1/N$. This leads to a proposal for a test of quantum supremacy. We will measure the quality of an algorithm A for a given number of qubits n as the difference between its cross entropy and the cross entropy of a uniform classical sampler. The algorithm A can be an experimental quantum implementation, or a classical algorithm implementation with *polynomial* or *exponential* cost as long as it is actually executed on an existing classical computer. We call this quantity the cross entropy difference:

$$\begin{aligned} \Delta \mathrm{H}(p_A) &\equiv \mathrm{H}_0 - \mathrm{H}(p_A, p_U) \\ &= \sum_j \left(\frac{1}{N} - p_A(x_j|U)\right) \log \frac{1}{p_U(x_j)} \; . \end{aligned} \tag{10}$$

The cross entropy difference measures how well algorithm $A(U)$ can predict the output of a (typical) quantum random circuit U. This quantity is unity for the ideal random circuit if the entropy of the output distribution is equal to the entropy of the Porter-Thomas distribution, and zero for the uniform distribution, see Eqs. (3) and (8).

In an experimental setting we describe the evolution of the density matrix

$$\rho_{\mathcal{K}} = \mathcal{K}_U(|\psi_0\rangle\langle\psi_0|) \tag{11}$$

with a superoperator $\mathcal{K}_U$ which corresponds to the circuit U and takes into account initialization, measurement and gate errors. We refer to the experimental implementation as $A_{\mathrm{exp}}(U)$ and associate with it the probability distribution $p_{\mathrm{exp}}(x_j|U) = \langle x_j| \rho_{\mathcal{K}} |x_j\rangle$ and sample S_{exp}. Consistent with Eq. (1), the experimental cross entropy difference is

$$\alpha \equiv \mathbb{E}_U[\Delta \mathrm{H}(p_{\mathrm{exp}})] \; .$$

Quantum supremacy is achieved, in practice, when

$$1 \geq \alpha > C \; , \tag{12}$$

where a lower bound for C (see also discussion below) is given by the performance of the best classical algorithm A^* known executed on an existing classical computer,

$$C = \mathbb{E}_U[\Delta \mathrm{H}(p^*)] \; . \tag{13}$$

Here p^* is the output distribution of A^*.

The space and time complexity of simulating a random circuit by using tensor contractions is exponential in the treewidth of the quantum circuit, which is proportional to $\min(d, n)$ in a 1D lattice, and $\min(d\sqrt{n}, n)$ in a 2D lattice [42]. For large depth d, algorithms are limited by the memory required to store the wavefunction in random-access memory, which in single precision is $2^n \times 2 \times 4$ bytes. For $n = 48$ qubits this requires at least 2.252 Petabytes, which is approximately the limit of what can be done on todays large-scale supercomputers.[4] For circuits of small depth or less than approximately 48 qubits, direct simulation is viable so $C = 1$ and quantum supremacy is impossible. Beyond this regime we are limited to an estimation of the Feynman path integral corresponding to the unitary transformation U. In this regime, the lower bound for C decreases exponentially with the number of gates $g \gg n$, see App. H.

We now address the question of how the cross entropy difference α can be estimated from an experimental sample of bit-strings S_{exp} obtained by measuring the output of $A_{\text{exp}}(U)$ after m realizations of the circuit. For a typical sample S_{exp}, the central limit theorem applied to Eq. (10) implies that

$$\alpha \simeq \mathrm{H}_0 - \frac{1}{m}\sum_{j=1}^{m} \log \frac{1}{p_U(x_j^{\text{exp}})} \ , \tag{14}$$

where H_0 is defined after Eq. (8). The statistical error in this equation, from the central limit theorem, goes like $\kappa/\sqrt{m}$, with $\kappa \simeq 1$. The experimental estimation would proceed as follows:

1. Select a random circuit U by sampling from an available universal set of one and two-qubit gates, subject to experimental layout constraints.

2. Take a sufficiently large sample $S_{\text{exp}} = \{x_1^{\text{exp}}, \dots, x_m^{\text{exp}}\}$ of bit-strings x in the computational basis ($m \sim 10^3 - 10^6$).

3. Compute the quantities $\log 1/p_U(x_j^{\text{exp}})$ with the aid of a sufficiently powerful classical computer.

4. Estimate α using Eq. (14).

For large enough circuits, the quantity $p_U(x_j^{\text{exp}})$ can no longer be obtained numerically. At this point, $C \simeq 0$, and supremacy can be achieved. Unfortunately, this also implies that α can no longer be measured directly. We argue that the observation of a close correspondence between experiment, numerics and theory would provide a reliable foundation from which to extrapolate α. The value of α can be extrapolated from circuits that can be simulated because they have either less qubits (direct simulation), mostly Clifford gates (stabilizer simulations) [44] or smaller depth (tensor contraction simulations) [42].

In practice, the necessary value of α in Eq. (12) to claim quantum supremacy will be limited not only by the lower bound on C in Eq. (13), but also by the number of measurements necessary to estimate α with high precision in Eq. (14), possible experimental biases among the different circuit types used to extrapolate α, and the precision in the agreement between theory and experiment. Next, we present a theoretical error model for $\mathcal{K}_U$ (see Eq. (11)) and the corresponding estimate of α that can be compared with experiments.

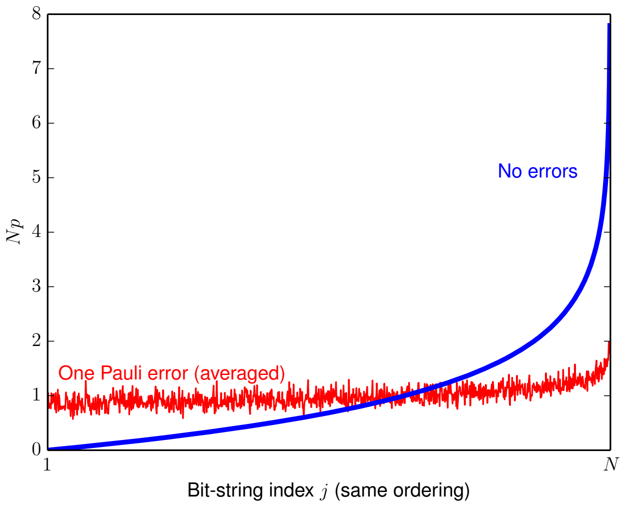

FIG. 3. The blue line shows the probabilities $p_U(x_j)$ of bit-strings x_j sorted in ascending order. The red line shows the corresponding probabilities after adding a Pauli error (X or Z) in a single location in the circuit, using the same ordering. The circuit used has 5×4 qubits and depth 25 (see Sec. IV). We average over all possible error locations. The average over errors gives almost the uniform distribution. The small residual correlation (slight upper curvature seen in the red line) is analyzed numerically in App. A.

III. FIDELITY ANALYSIS

The output $\rho_{\mathcal{K}}$ of the experimental realization $\mathcal{K}_U$ of a random circuit U is

$$\rho_{\mathcal{K}} = \tilde{\alpha} U \left|\psi_0\right\rangle\!\left\langle\psi_0\right| U^\dagger + (1-\tilde{\alpha})\sigma_U \ , \tag{15}$$

where $\left\langle\psi_0\right| U^\dagger \sigma_U U \left|\psi_0\right\rangle = 0$ and $\tilde{\alpha}$ is the circuit fidelity. The matrix σ_U represents the effect of errors. Because U is a random circuit implementing a chaotic evolution, we assume that the probabilities $p_U(x)$ and $\left\langle x\right| \sigma_U \left|x\right\rangle$ are almost uncorrelated. This is supported by numerical simulations, see Fig. 3 and App. A. Under this ansatz, by the same arguments leading to Eq. (8), we obtain that the circuit fidelity $\tilde{\alpha}$ is approximately equal to the cross entropy difference α

$$\alpha = \mathbb{E}_U[\Delta \mathrm{H}(p_{\text{exp}})] \approx \tilde{\alpha} \ . \tag{16}$$

Estimating the circuit fidelity by directly measuring the cross entropy (see Eq. (14)) is a fundamentally new way to characterize complex quantum circuits.

[4] Trinity, the sixth fastest supercomputer in TOP500 [43], has ∼ 2 Petabytes of main memory - one of the largest among existing supercomputers today.

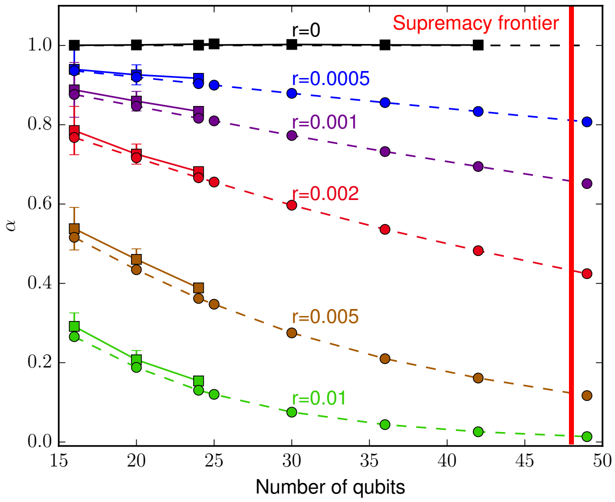

FIG. 4. The circuit fidelity α as a function of the number of qubits. Different colors correspond to different Pauli error rates $r_2 = r_{\text{init}} = r_{\text{mes}} = r$ and $r_1 = r/10$. Circular markers correspond to the numerically simulated fidelities, Eq. (17). Square markers correspond to the average cross entropy difference among 100 instances, Eq. (10). The circuit depth in these simulations is 25 (see Sec. IV). The red line, at 48 qubits, is a reasonable estimate of the largest size that can be simulated with state-of-the-art classical supercomputers in practice. Using state-of-the-art superconducting circuits we expect $\alpha \gtrsim 0.1$ for a 7×7 circuit. Error bars correspond to the standard deviation among 100 instances.

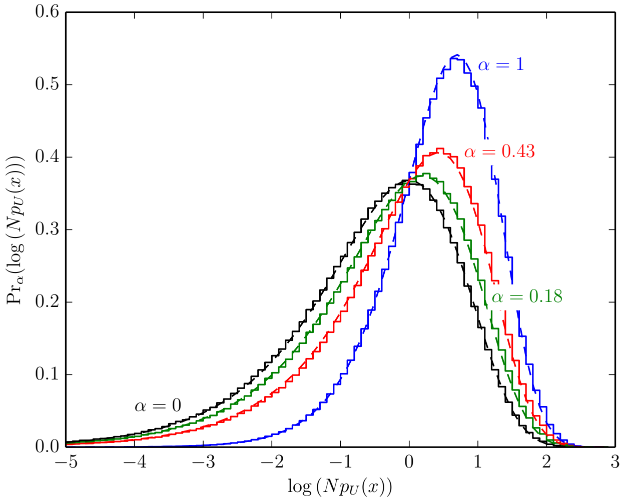

FIG. 5. Probability distribution of $\log(Np_U(x))$ where bit-strings x are sampled from a circuit of fidelity α. The continuous step histograms are obtained from numerical simulations with different Pauli error rates $r_2 = r_{\text{init}} = r_{\text{mes}} = r$ and $r_1 = r/10$. The values of r are $r = 0$ for $\alpha = 1$ (blue), $r = 0.005$ for $\alpha = 0.43$ (red), $r = 0.01$ for $\alpha = 0.18$ (green) and uniform sampling of bit-strings for $\alpha = 0$. The value of α is estimated using Eq. (17). The superimposed dashed lines correspond to the theoretical distribution of Eq. (19). We chose a circuit of 5×4 qubits and depth 25 (see Sec. IV).

The standard approach to studying circuit fidelity is the digital error model where each quantum gate is followed by an error channel [45, 46]. Within this model, the circuit fidelity can be estimated as [45, 47]

$$\alpha \approx \exp(-r_1 g_1 - r_2 g_2 - r_{\text{init}} n - r_{\text{mes}} n) , \qquad (17)$$

where $r_1, r_2 \ll 1$ are the Pauli error rates for one and two-qubit gates, $r_{\text{init}}, r_{\text{mes}} \ll 1$ are the initialization and measurement error rates, and $g_1, g_2 \gg 1$ are the numbers of one and two-qubit gates respectively.

We have performed numerical simulations of random circuits in the presence of errors by introducing a depolarizing channel after each gate [31, 32, 45, 46, 48–51] (see Sec. IV for details about the circuits design). Errors in the depolarizing channel after each two-qubit gate are emulated by applying one of the 15 possible combinations of products of two Pauli operators (excluding the identity) with an equal probability of $r_2/15$. Similarly, we apply a randomly selected single Pauli matrix after each one-qubit gate with an equal probability of $r_1/3$. Initialization and measurement errors are simulated by applying a bit-flip with probability r_{init} and r_{mes} respectively. Figure 4 shows the cross entropy difference, Eq. (10), obtained from these simulations, and the estimated fidelity, Eq. (17). We observe a good agreement between these two quantities. The small difference between the cross entropy difference and the estimated fidelity is due to residual correlations analyzed numerically in App. A.

Note that the cross entropy difference of the ideal circuit ($r = 0$ in the figure) is almost exactly one, indicating that at this depth all sizes studied are in the Porter-Thomas regime. Details of the optimizations employed for the simulation of the larger circuits, of up to 42 qubits, are given in App. B. These are the largest quantum circuits simulated to-date for a computational task that approaches quantum supremacy.

Because chaotic states are maximally entangled [17], even one Pauli error completely destroys the state [52], as seen in numerical data in Fig. 3. More formally, consider a sequence of arbitrary quantum channels interleaved with unitaries randomly chosen from a group that is also a 2-design. This is equivalent to a sequence of channels with the same average fidelity in which all the channels (except the last one) are transformed into depolarizing channels [49, 51]. Although individual two-qubit gates are not a 2-design for n qubits, a large part of the evolution of a typical random circuit takes place in the Porter-Thomas regime. We therefore make the following ansatz for the output state $\rho_{\mathcal{K}}$

$$\rho_{\mathcal{K}} = \alpha \left|\psi_d\right\rangle \left\langle\psi_d\right| + (1-\alpha)\frac{\mathbb{1}}{N} . \qquad (18)$$

As seen in Fig. 2, errors alter the shape of the Porter-Thomas distribution, approaching the uniform distribu-

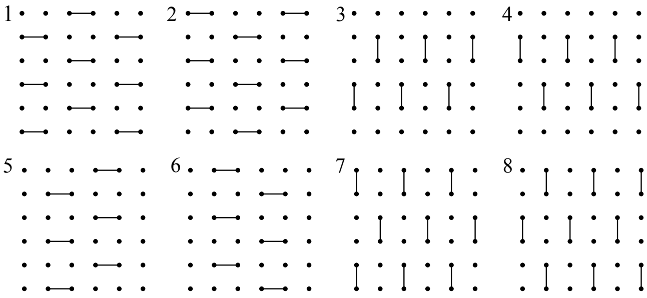

FIG. 6. Layouts of CZ gates in a 6 × 6 qubit lattice. It is currently not possible to perform two CZ gates simultaneously in two neighboring superconducting qubits [31, 32, 45, 48]. We iterate over these arrangements sequentially, from 1 to 8.

tion as $\alpha \to 0$.

The cross entropy difference $\Delta\mathrm{H}$ defined in Eq. (10) is given by the probability distribution of $\log(p_U(x))$ where the bit-strings x are sampled from the output $\rho_{\mathcal{K}}$ of a circuit implementation with fidelity α. Using Eq. (18) and the Porter-Thomas distribution for $p_U(x)$ we obtain

$$\mathrm{Pr}_\alpha(z) = e^{z-e^z}\left(1-\alpha\left(e^z-1\right)\right) \ , \tag{19}$$

where $z = \log(Np)$. If bit-strings are sampled uniformly, $-\log p_U(x)$ has a Gumbel distribution. We find a good fit between this expression and numerical simulations, see Fig. 5. The value of α corresponding to a given Pauli error rate per gate can be estimated using Eq. (17).

IV. CONVERGENCE TO PORTER-THOMAS

In this section we report the results of numerical simulations on the required depth to approximate the Porter-Thomas distribution using planar quantum circuits that would be feasible to implement using state-of-the-art superconducting qubit platforms [31, 32, 45, 48]. The following circuits were chosen through numerical optimizations to minimize the convergence time to Porter-Thomas.

1. Start with a cycle of Hadamard gates (0 clock cycle).

2. Repeat for d clock cycles:

   (a) Place controlled-phase (CZ) gates alternating between eight configurations similar to Fig. 6.

   (b) Place single-qubit gates chosen at random from the set $\{\mathrm{X}^{1/2}, \mathrm{Y}^{1/2}, \mathrm{T}\}$ at all qubits that are not occupied by the CZ gates at the same cycle (subject to the restrictions below). The gate $\mathrm{X}^{1/2}$ ($\mathrm{Y}^{1/2}$) is a $\pi/2$ rotation around the X (Y) axis of the Bloch sphere, and the non-Clifford T gate is the diagonal matrix $\{0, e^{i\pi/4}\}$.

In addition, single-qubit gates are placed subject to the following rules:

- Place a gate at qubit q only if this qubit is occupied by a CZ gate in the previous cycle.
- Place a T gate at qubit q if there are no single-qubit gates in the previous cycles at qubit q except for the initial cycle of Hadamard gates.
- Any gate at qubit q should be different from the gate at qubit q in the previous cycle.

In the numerical study we calculate statistics corresponding to measurements in the computational (or Z) basis after each cycle. Because the CZ gates are diagonal in this basis, some gates before the measurement could be simplified away. The circuit would be harder to simplify if a cycle of Hadamards is applied before measuring in the Z basis. We did not apply a final cycle of Hadamards in the numerical study because it would double the computational run time, as the cycle of Hadamards would have to be undone after collecting statistics at cycle t before moving to cycle $t+1$. We argue that the Porter-Thomas form of the output distribution, characteristic of chaotic systems, makes it unlikely that these circuits can be simplified substantially (see also Secs. I and V).

We estimate numerically the corresponding treewidth for the tensor network contraction algorithm applied to these circuits [42]. The treewidth of circuits with 7 × 6 qubits becomes intractably large for tensor contractions when the gate depth exceeds approximately 25 for realistic circuits, see App. C. The reason why this depth is larger than the length of a side of the circuit lattice is because 8 cycles are needed to complete a single 2D lattice, see Fig. 6.

Random circuits approximate a pseudo-random distribution [53, 54] with logarithmic depth in a fully connected architecture [12, 13, 15]. These circuits can be embedded with depth proportional to $\sqrt{n}$, up to polylogarithmic factors in n, in a 2D lattice [55]. Consistent with our earlier discussion, we study how the entropy of the circuit output converges to the entropy of the Porter-Thomas distribution, Eq. (3). Figure 4 ($r = 0$ line) shows that for all sizes of circuits up to 7×6 qubits, constructed according to the restrictions given above, our simulations reveal that the output distribution has the same entropy as the Porter-Thomas distribution. Figure 7 shows the output distribution entropy as a function of circuit depth. Circuits approach the Porter-Thomas regime with approximately ten cycles. Note that the initial entropy corresponds to the uniform distribution due to the first layer of Hadamards. Gates in the first cycles are diagonal and do not change the output entropy.

To develop intuition about the chaotic evolution of the wavefunction, we focus on the degree of delocalization of the distribution $p_U(x_j)$. The degree of delocalization is captured by the moments of the distribution $\mathrm{PR}_t^{(k)} = \sum_j |\langle x_j|\psi_t\rangle|^{2k}$, usually referred to as participation ratios [56, 57]. If the wavefunction has support over

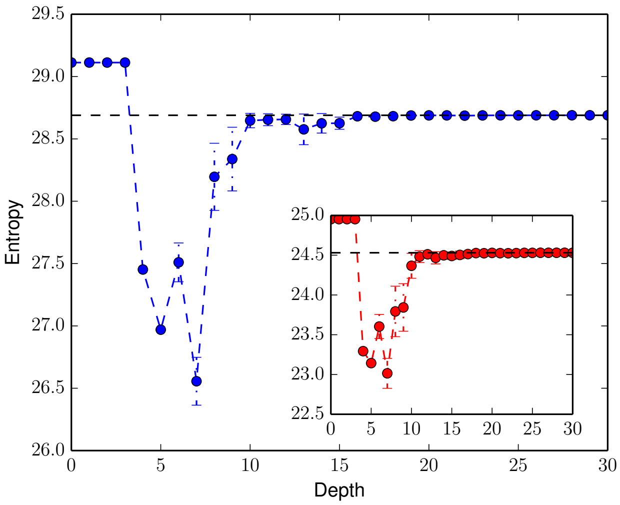

FIG. 7. Mean entropy of the output distribution as a function of depth. The main figure pertains to circuits with 7×6 qubits, and the inset pertains to circuits with 6×6 qubits. The black dashed lines correspond to the entropy of the Porter-Thomas distribution. Error bars are standard deviations among different circuit instances.

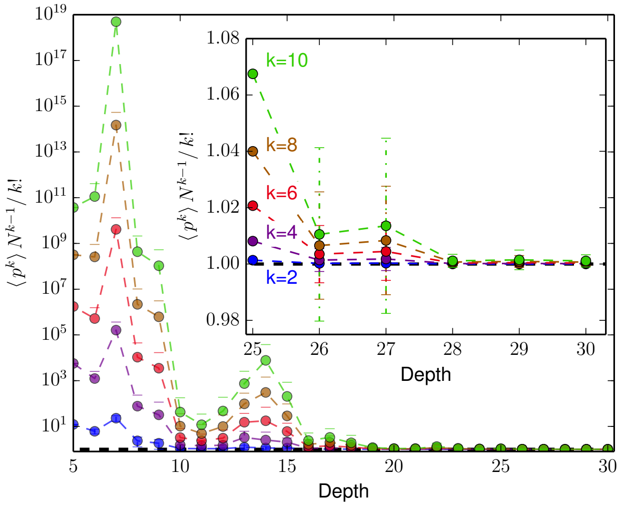

FIG. 8. Mean normalized moments $k \in [2, .., 10]$ of the output distribution as a function of depth for circuits with 7×6 qubits. The black dashed line at the bottom corresponds to the Porter-Thomas distribution. Error bars correspond to the standard deviation between different circuit instances.

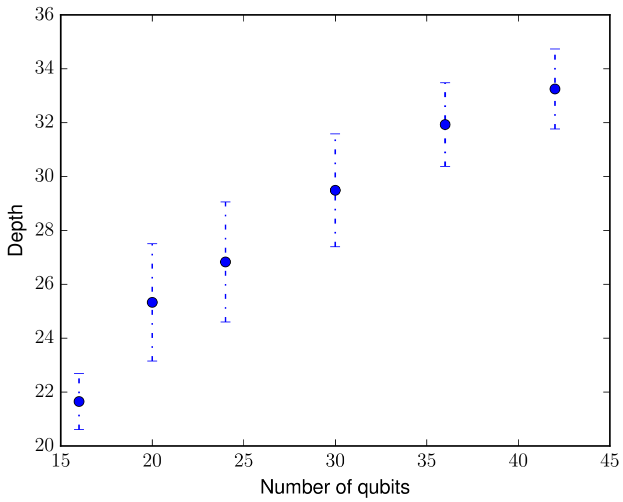

FIG. 9. First cycle in a random circuit instance such that the entropy remains within 4-sigma of the Porter-Thomas entropy during all the following cycles. Markers show the mean among instances and error bars correspond to the standard deviation among circuit instances.

$\xi_t N$ local basis vectors, then $\mathrm{PR}_t^{(k)} \propto N^{-k+1}\xi_t^{-k}$. As t increases, $\xi_t \to 1$ and the wavefunction becomes a pseudo-random vector sampled uniformly from Hilbert space. At that point, finite moments of the distribution converge to Porter-Thomas, $\mathrm{PR}_t^{(k)} \to N^{-k+1}k!$ [11, 12, 14]. Importantly, we find numerically that convergence is achieved for all finite moments at a similar depth. This is evidenced in Fig. 8 for moments up to $k = 10$ with circuits consisting of 7×6 qubits.

We also studied the expected convergence to Porter-Thomas with depth proportional to $\sqrt{n}$ using a stronger criterion. The standard deviation of the entropy between different quantum states drawn from the Porter-Thomas distribution scales as $\approx 0.75 \cdot 2^{-n/2}$. In Fig. 9 we show the first cycle of each random circuit instance for which the entropy remains within 4-sigma of the Porter-Thomas entropy during all the following cycles. These data indicates that the required depth to achieve this criteria grows sublinearly in n. We show a similar plot for circuits with denser layouts of CZ gates, which can be more appropriate for other qubit implementations, in App. E.

V. COMPUTATIONAL HARDNESS OF THE CLASSICAL SAMPLING PROBLEM

The distribution $p_U(x) \propto 1/2^n$ is highly delocalized in the computational basis and in any basis obtained from local rotations of the computational basis. Therefore, it is impossible to estimate $p_U(x)$ for any x, even using a quantum computer, as doing so would require an exponential number of measurements. Nevertheless, the distribution $p_U(x)$ can be sampled efficiently by performing measurements on the state produced by the shallow random circuit U on a quantum computer. In contrast, as we argued above from the chaotic nature of the evolution, a classical algorithm can only sample from the distribution $p_U(x)$ if it can compute this function explicitly. This requires resources which grow exponentially in n, making the problem intractable even for modest sized random quantum circuits.

This intuitive argument can be made more rigorous in the asymptotic limit using computational complexity theory. Previous studies have introduced related sampling problems that a quantum computer can solve without having the ability to estimate $p_U(x)$ [18–22, 24–28]. In this section we will extend the method used to show the computational hardness of sampling commuting random circuits (IQP) [21, 27] to the general case of universal random circuits.

We will first describe the computational complexity class of estimating a probability $p_{\rm pcl}(x)$ of a polynomial classical sampling algorithm. This is based on the fact that a random classical algorithm uses random bits, which is very different from the intrinsic randomness of quantum mechanics. We will then argue that approximating $p_U(x)$ belongs to a much harder complexity class, which implies that there does not exist an efficient classical sampling algorithm.

A. General overview of the computational complexity argument

A classical sampling algorithm corresponds to the evaluation of a function

$$f(w,y) = x \ . \tag{20}$$

Here the bit-string $w = \{w_1, \dots, w_k\}$ encodes the problem instance, y is a vector of random bits $y = \{y_1, \dots, y_\ell\}$ chosen uniformly and x is the output bit-string. For fixed w and x, the number W_x of solution vectors y of Eq. (20) defines the probability $q(x) = W_x/2^\ell$ of getting a sample x. Assume that evaluating the function f can be done in a time which scales polynomially in the number of input bits $k+\ell$, with ℓ polynomial in k. Then, the problem of determining if there is a solution vector y to Eq. (20) with fixed w and x belongs to the complexity class NP. A complexity theory abstraction that solves this general problem is called an *NP-oracle*. An important result in computer science, the so-called Stockmeyer Counting Theorem [58], states that probabilistically approximating the number of solutions W_x, and therefore $q(x)$, to within a multiplicative factor, can also be performed with an NP oracle, see App. F.

A classical sampling algorithm simulating a quantum random circuit U must output bit-strings x with probability $q(x)$ approximating $p_U(x)$. The input vector w to the corresponding function $f(w,y)$ is a description of the circuit U, which is polynomial in the number of qubits n. It has been shown that, in the case of commuting quantum circuits, the function $p_U(x) = |\langle x|\psi\rangle|^2$ encodes the partition function of a random complex Ising model [21, 27]

$$\langle x|\psi\rangle = \lambda \sum_s e^{i\theta H_x(s)} \ , \quad H_x(s) = h_x\cdot s + s\cdot\hat{J}\cdot s \ , \tag{21}$$

where $H_x(s)$ is a classical energy, s is a vector of classical spins ±1, h_x is a vector of local fields, $\hat{J}$ is the coupling matrix, $i\theta$ is the inverse imaginary temperature and λ is a scaling constant. The partition function can also be written as $\sum_j M_j e^{i\theta E_j}$ where M_j is the number of solutions s to the equation $H_x(s) = E_j$. In general, the M_j's grow exponentially in the number of classical spins.

The partition function at low *real-valued* temperatures T (with $\theta = i/T$) is hard to approximate only because the sum in Eq. (21) is dominated by low energy states. The Stockmeyer Counting Theorem implies that probabilistically approximating the corresponding M_j within a multiplicative error can be done with an NP-oracle, because for any given s the energy $H_x(s)$ can be calculated efficiently. This results in a multiplicative error estimation of the partition function. In contrast, for purely imaginary temperatures i/θ, the sum $\sum_j M_j e^{i\theta E_j}$ is determined by the intricate cancellations between individual terms, each exponentially large in magnitude. A discussion of this cancellation for the case of random circuits is given in the next subsection. An approximation of M_j with multiplicative error is not sufficient to estimate the partition function. Therefore, the case with purely imaginary temperatures is much harder than the real-valued case.

These intuitive arguments are supported by the strongly held conjecture in computational complexity theory that probabilistically approximating partition functions with purely imaginary temperatures is much harder, in the worst case, than any problem which can be solved with an NP oracle [21, 23, 59]. Reference [27] argues that because random instances of Ising models have no structure making them easier, the same conjecture applies to any sufficiently large fraction of partition functions of random complex Ising models.

Assume now that there exists an approximate classical sampling algorithm for the distribution p_U with asymptotic complexity polynomial in n and small distance in the ℓ_1 norm. From the convergence of the second moment of p_U to the Porter-Thomas distribution found numerically, it would then follow from the proof in Ref. [27] that a fraction of these probabilities could be probabilistically approximated with multiplicative error using an NP-oracle, see App. G. As argued above, this is implausible for a complex partition function with the general form of Eq. (21). We will show in the next section that $p_U(x)$ can be mapped directly to the partition function of a quasi three-dimensional random Ising model, with no apparent structure that makes it easier to approximate than a random instance. If we conjecture that a sufficient large fraction of these instances is as hard to approximate as the worst case, we must conclude that such an efficient classical sampling cannot be achieved.

B. The partition function for random circuits

While our approach for mapping circuits to partition functions can be applied to any circuit, we focus here on the particular case of a quantum circuit U as described in Sec. IV. Known algorithms for mapping universal quantum circuits to partition functions of complex Ising models use polynomial reductions to a universal gate set [60]. Here we provide a direct construction, which allows us to define a random ensemble of Ising models without apparent structure. We represent the circuit by a product of unitary matrices $U^{(t)}$ corresponding to different clock cycles t, with the 0-th cycle formed by Hadamard gates. We introduce the following notation for the amplitude of a particular bit-string after the final cycle of the circuit,

$$\langle x|\psi_d\rangle = \sum_{\{\sigma^t\}} \prod_{t=0}^{d} \langle \sigma^t| \, U^{(t)} \, |\sigma^{t-1}\rangle \,, \quad |\sigma^d\rangle = |x\rangle \,. \qquad (22)$$

Here $|\sigma^t\rangle = \otimes_{j=1}^{n} |\sigma_j^t\rangle$ and the assignments $\sigma_j^t = \pm 1$ correspond to the states $|0\rangle$ and $|1\rangle$ of the j-th qubit, respectively. The expression (22) can be viewed as a Feynman path integral with individual paths $\{\sigma^{-1}, \sigma^0, \ldots, \sigma^d\}$ formed by a sequence of the computational basis states of the n-qubit system. The initial condition for each path corresponds to $\sigma_j^{-1} = 0$ for all qubits and the final point corresponds to $|\sigma^d\rangle = |x\rangle$.

Assuming that a T gate is applied to qubit j at the cycle t, the indices of the matrix $\langle \sigma^t| \, U^{(t)} \, |\sigma^{t-1}\rangle$ will be equal to each other, i.e. $\sigma_j^t = \sigma_j^{t-1}$. A similar property applies to the CZ gate as well. The state of a qubit can only flip under the action of the gates H, $\mathrm{X}^{1/2}$ or $\mathrm{Y}^{1/2}$. We refer to these as two-sparse gates as they contain two nonzero elements in each row and column (unlike T and CZ). This observation allows us to rewrite the path integral representation in a more economic fashion.

Through the circuit, each qubit j has a sequence of two-sparse gates applied to it. We denote the length of this sequence as $d(j)+1$ (this includes the 0-th cycle formed by a layer of Hadamard gates applied to each qubit). In a given path the qubit j goes through the sequence of spin states $\{s_j^k\}_{k=0}^{d(j)}$, where, as before, we have $s_j^k = \pm 1$. The value of s_j^k in the sequence determines the state of the qubit *immediately after* the action of the k-th two-sparse gate. The last element in the sequence is fixed by the assignment of bits in the bit-string x,

$$s_j^{d(j)} = x^{(j)} \,, \quad j \in [1 \,..\, n] \,. \qquad (23)$$

Therefore, an individual path in the path integral can be encoded by the set of $G = \sum_{j=1}^{n} d(j)$ binary variables $s = \{s_j^k\}$ with $j \in [1 \,..\, n]$ and $k \in [0 \,..\, d(j)-1]$. One can easily see from the explicit form of the two-sparse gates that the absolute values of the probability amplitudes associated with different paths are all the same and equal to $2^{-G/2}$. Using this fact we write the path integral (22) in the following form

$$\langle x|\psi_d\rangle = 2^{-G/2} \sum_{s} \exp\left(\frac{i\pi}{4} H_s(x)\right) \,. \qquad (24)$$

Here $\exp(i\pi H_s(x)/4)$ is a phase factor associated with each path that depends explicitly on the end-point condition (23).

The value of the phase $\pi H_s/4$ is accumulated as a sum of discrete phase changes that are associated with individual gates. For the k-th two-sparse gate applied to qubit j we introduce the coefficient α_j^k such that $\alpha_j^k = 1$ if the gate is $\mathrm{X}^{1/2}$ and $\alpha_j^k = 0$ if the gate is $\mathrm{Y}^{1/2}$. Thus, the total phase change accumulated from the application of $\mathrm{X}^{1/2}$ and $\mathrm{Y}^{1/2}$ gates equals

$$\frac{i\pi}{4} H_s^{\mathrm{X}^{1/2}}(x) = \frac{i\pi}{2} \sum_{j=1}^{n} \sum_{k=0}^{d(j)} \alpha_j^k \frac{1 + s_j^{k-1} s_j^k}{2} \,, \qquad (25)$$

$$\frac{i\pi}{4} H_s^{\mathrm{Y}^{1/2}}(x) = i\pi \sum_{j=1}^{n} \sum_{k=0}^{d(j)} (1 - \alpha_j^k) \frac{1 - s_j^{k-1}}{2} \frac{1 + s_j^k}{2} \,.$$

As mentioned above, the dependence on x arises due to the boundary condition (23). Note that we have omitted constant phase terms that do not depend on the path s.

We now describe the phase change from the action of gates T and CZ. We introduce coefficients $d(j,t)$ equal to the number of two-sparse gates applied to qubit j over the first t cycles (including the 0-th cycle of Hadamard gates). We also introduce coefficients τ_j^t such that $\tau_j^t = 1$ if a T gate is applied at cycle t to qubit j and $\tau_j^t = 0$ otherwise. Then the total phase accumulated from the action of the T gates equals

$$\frac{i\pi}{4} H_s^{T}(x) = \frac{i\pi}{4} \sum_{j=1}^{n} \sum_{t=0}^{d} \tau_j^t \frac{1 - s_j^{d(j,t)}}{2} \,. \qquad (26)$$

For a given pair of qubits (i,j), we introduce coefficients z_{ij}^t such that $z_{ij}^t = 1$ if a CZ gate is applied to the qubit pair during cycle t and $z_{ij}^t = 0$ otherwise. The total phase accumulated from the action of the CZ gates equals

$$\frac{i\pi}{4} H_s^{\mathrm{CZ}}(x) = i\pi \sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{t=0}^{d} z_{ij}^t \frac{1 - s_i^{d(i,t)}}{2} \frac{1 - s_j^{d(j,t)}}{2} . \qquad (27)$$

One can see from comparing (24) with (25)-(27) that the wavefunction amplitudes $\langle x|\psi_d\rangle$ take the form of a partition function of a classical Ising model with energy H_s for a state s and purely imaginary inverse temperature $i\pi/4$. The total phase for each path takes 8 distinct values (mod 2π) equal to $[0, \pi/4 \,..\, 7\pi/4]$. The function $H_s(x)$ can be written as a sum of three different types of terms

$$H_s(x) = H_s^{(0)} + H_s^{(1)} + H^{(2)} \,. \qquad (28)$$

Here

$$H_s^{(0)} = \sum_{i=1}^{n} \sum_{k=1}^{d(i)-1} h_i s_i + \sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{k=1}^{d(i)-1} \sum_{l=1}^{d(j)-1} \mathcal{J}_{ij}^{kl} s_i^k s_j^l . \qquad (29)$$

is the energy term quadratic in spin variables and expressed in terms of the Ising coupling coefficients $\mathcal{J}_{ij}^{kl}$ and local fields h_i^k to be given below. It does not depend on the spin configuration x of the final point on the paths. $H_s^{(1)}$ is a bilinear function of Ising spin variables s and x

$$H_s^{(1)}(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{d(i)-1} b_{ij}^k s_i^k x^{(j)} \ . \qquad (30)$$

The term $H^{(2)}(x)$ depends on x but not s. For brevity, we do not provide its explicit form.

The local fields h_j are computed as

$$h_i^k = \alpha_i^{k+1} - \alpha_i^k - \frac{1}{2} J_i^k - \sum_{j=1}^{n} \sum_{l=1}^{d(j)} J_{ij}^{k\,l} \qquad (31)$$

and the coupling constants $\mathcal{J}_{ij}^{kl}$ equal

$$\mathcal{J}_{ij}^{kl} = J_{ij}^{kl} + \frac{1}{2} \delta_{i,j} (\delta_{k-1,l} + \delta_{k,l-1}) \left(2\alpha_i^{(k+l+1)/2} - 1 \right) \qquad (32)$$

where

$$J_{ij}^{kl} = \sum_{t=1}^{d} \delta_{k,d(i,t)} \delta_{l,d(j,t)} z_{ij}^t \ , \qquad (33)$$

and

$$J_i^k = \sum_{t=1}^{d} \delta_{k,d(i,t)} \tau_i^t \ . \qquad (34)$$

The coupling coefficients b_{ij}^k in (30) equal

$$b_{ij}^k = \delta_{k,d(i)-1} \delta_{ij} (2\alpha_j^{d(j)} - 1) + J_{ij}^{kd(j)} \ . \qquad (35)$$

The Ising coupling for spin $s_j^{d(j)} = x^{(j)}$ induces an additional local field $\sum_{j=1}^{n} \sum_{k=1}^{d(i)-1} b_{ij}^k x^{(j)}$ on spin s_i^k as shown in (29).

To understand the structure of the graph defined by the Ising couplings (32) we study the statistical ensemble of $\mathcal{J}_{ij}^{kl}$. For simplicity, we will analyze circuits composed of d layers, each layer consisting of a cycle of single-qubit gates followed by a cycle of two-qubit CZ gates (see App. E). We also assume here that the layout of the two-qubit CZ gates is random, and that in the single-qubit gate cycles the gates $X^{1/2}$, $Y^{1/2}$, and T are applied to a qubit with equal probabilities.

To describe the evolution of qubit states under the action of the gates we need to introduce a third dimension to describe the graph of the Ising couplings, Eq. (32). For each qubit j we introduce a "worldline" with a grid of points enumerated by $t \in [1 \ .. \ d]$, each corresponding to a layer. We denote the layer numbers where the function $d(j,t)$ increases from $k-1$ to k by a two-sparse gate applied to qubit j as t_j^k. We associate Ising spins $\{s_j^k\}_{k=0}^{d(j)-1}$ to vertices of the graph located at the grid points $\{t_j^k\}$ along the worldline j.

Consider a pair of vertices corresponding to spins s_i^k and s_j^l associated with the two adjacent qubits i and j. Then the coefficient J_{ij}^{kl} equals to the number of applied CZ gates that couple qubits i an j during the sequence of layers $[\max(t_i^k, t_j^l) \ .. \ (\min(t_i^{k+1}, t_j^{l+1}) - 1)]$. The distribution of J_{ij}^{kl} can be written in the following form

$$\Pr[J_{ij}^{kl} = r] \equiv P(r) = \sum_{q=0}^{\infty} p(r|q) p(q) \, , \qquad (36)$$

Here $p(q) = \frac{8}{9} \left(\frac{1}{3}\right)^{2q}$ is the probability of having *no* two-sparse gates applied to qubits i and j for q layers and then having a two-sparse gate applied to at least one of them in the $(q+1)^{\text{st}}$ layer. Also $p(r|q) = \binom{q+1}{r} p_{\text{CZ}}^r (1 - p_{\text{CZ}})^{q+1-r}$ is the probability of having r CZ gates over $q+1$ layers applied between a given pair of neighboring qubits. Finally, we have for $P(r)$

$$P(r) = \begin{cases} \dfrac{1 - p_{\text{CZ}}}{1 + p_{\text{CZ}}/8}, & r = 0 \\ \dfrac{9}{1 + p_{\text{CZ}}/8} \left(\dfrac{p_{\text{CZ}}/8}{1 + p_{\text{CZ}}/8} \right)^r & r > 0 \ . \end{cases} \qquad (37)$$

For a square grid of qubits $p_{\text{CZ}} \simeq 1/4$. One can see from (37) that for $r \geq 1$ the distribution $\Pr[J_{ij}^{kl} = r]$ decays exponentially with r and $P(r+1)/P(r) \simeq p_{\text{CZ}}/8 \simeq 1/32$. Therefore, the most likely values of J_{ij}^{kl} are 0, corresponding to the probability $P(0) \simeq 1 - p_{\text{CZ}}$, and 1, corresponding to the probability $P(1) \simeq 9p_{\text{CZ}}/8$. The high probability of having no traversal couplings between qubits relates to the comparatively slow growth of the circuit graph treewidth, see App. C.

For fixed qubit indexes (i,j), it is of interest to derive the conditional distribution $\mathfrak{p}(l|k)$ for spin s_i^k to couple to spin s_j^l. To obtain it we first introduce the probability $\mathfrak{p}_k(t)$ corresponding to the condition $t_i^k = t$ of having the k-th vertex located exactly at the layer t of a given wordline. Not too close to the end of the circuit ($d - t \gg \sqrt{d}$) we have

$$\mathfrak{p}_k(t) = \binom{t-1}{k-1} \left(\frac{1}{3}\right)^{t-k} \left(\frac{2}{3}\right)^k , \quad \sum_{t=k}^{\infty} \mathfrak{p}_k(t) = 1, \qquad (38)$$

Similarly, the probability $\mathfrak{p}^t(l)$ of having exactly l vertices located within t layers of a given wordline ($t_j^l \leq t$) equals

$$\mathfrak{p}^t(l) = \binom{t}{l}\left(\frac{1}{3}\right)^{t-l}\left(\frac{2}{3}\right)^{l}\,, \quad \sum_{l=0}^{t}\mathfrak{p}^t(l) = 1\;. \tag{39}$$

The above conditional distribution $\mathfrak{p}(l|k)$ of the values of l given k equals

$$\mathfrak{p}(l|k) = \sum_{t}\mathfrak{p}^t(l)\mathfrak{p}_k(t)\;. \tag{40}$$

Approximating the binomial coefficients with the Stirling formula we obtain

$$\mathfrak{p}(l|k) \simeq \sqrt{\frac{3}{2\pi(k+l)}}\exp\left(-\frac{3(k-l)^2}{2(k+l)}\right)\;. \tag{41}$$

The above equation is asymptotically correct for k, l not to close to the start and end points of the circuit, and $|k-l| \ll d$.

In summary, the coupling graph corresponding to the coefficients $\mathcal{J}_{ij}^{kl}$ represents a quasi three-dimensional structure formed by wordlines corresponding to qubits located on a 2D lattice. According to (32), in the same worldline only neighboring vertices are coupled. The strength of the coupling is $\pm 1/2$ depending on the type of the two-sparse gate. In general, each vertex can be "laterally" coupled to other vertices located on the neighboring wordlines. The probability distribution of the coupling coefficients has exponential form, Eq. (37). Differences between the vertex indices that are involved in the lateral couplings obey a local Gaussian distribution, Eq. (41).

Finally, note that Eq. (24) can be written in the form $\langle x|\psi_d\rangle = 2^{-G/2}Z$, where $Z = \sum_{j=0}^{7} M_j e^{i\frac{2\pi}{8}E_j}$ is a partition function, the E_j's are different energies of the Ising model (mod 8) and $M_j \sim 2^G$. Furthermore, for a delocalized state $|\langle x|\psi\rangle| \sim 2^{-n/2}$. Therefore, the partition function $|Z| \sim 2^{(G-n)/2}$ is exponentially smaller in G than the individual terms M_j in its sum. This very strong cancellation prevents any efficient algorithm from being able to accurately estimate the quantity $\langle x|\psi\rangle$ (see also App. H).

Note that if a quantum circuit uses only Clifford gates (not T gates), the total phase for each spin configuration in the partition function (mod 2π) is restricted to $[0, \pi/2, \pi, 3\pi/2]$. In these case, the corresponding partition function can be calculated efficiently [23, 59, 61].

VI. CONCLUSION

In the near future, quantum computers without error correction will be able to approximately sample the output of random quantum circuits which state-of-the-art classical computers cannot simulate [18–28]. We have introduced a well-defined metric for this computational task. If an experimental quantum device achieves a cross entropy difference surpassing the performance of the state-of-the-art classical competition, this will be a first demonstration of quantum supremacy [29]. The cross entropy can be measured up to the quantum supremacy frontier with the help of supercomputers. After that point it can be extrapolated by varying the number of qubits, the number of non Clifford gates [44], and/or the circuit depth [42]. Furthermore, the cross entropy can be approximated independently from estimates of the circuit fidelity. Quantum supremacy can be claimed if the theoretical estimates are in good agreement with the experimental extrapolations.

A crucial aspect of a near-term quantum supremacy proposal is that the computational task can only be performed classically through a direct simulation with cost exponential in the number of qubits. Direct simulations are required for chaotic systems, such as random quantum circuits [5, 7]. A simulation can be done in several ways: evolving the full wavefunction; calculating matrix elements of the circuit unitary with tensor contractions [42]; using the stabilizer formalism [44]; or summing a significant fraction of the corresponding Feynman paths in the partition function of an Ising model with imaginary temperature, see App. H. We study the cost of all these algorithms and conclude that, with state-of-the-art supercomputers, they fail for universal random circuits with more than approximately 48 qubits and depth 25.

We related the computational hardness of this problem, originating from the chaotic evolution of the wavefunction, to the sign problem emerging from the cancellation of exponentially large terms in a partition function of an Ising model with imaginary temperature. This finding is made more rigorous by results in computational complexity theory [21, 23, 27, 59]. Following previous works [22, 27, 62–64], we argue that, under certain assumptions, there does not exist an efficient classical algorithm which can sample the output of a random quantum circuit with a constant error (in the ℓ_1 norm) in the limit of a large number of qubits n (see Eq. (G2)). Unfortunately, achieving a constant error in the limit of large n requires a fault tolerant quantum computer, which will not be available in the near term [65, 66]. Nonetheless, it has been argued, also using computational complexity theory, that the *exact* output distribution of certain quantum circuits with a constant probability of error per gate is also asymptotically hard to simulate classically [67].

A specific figure of merit for a well defined computational task, naturally related to fidelity, as well as an accurate error model, are equally crucial for establishing quantum supremacy in the near-term. This is absent from previous experimental results with quantum systems which can not be simulated directly [68–74]. Without this, it is not clear if divergences between the experimental data and classical numerical methods [68, 72] are due to the effect of noise or other unaccounted sources. Furthermore, we note that the numerical simulation and experimental curves in Ref. [68] are reasonably well fitted by a rescaled cosine. Therefore, these curves can be approximately extrapolated efficiently classically.

Finally, the problem of sampling from the output distribution defined by a random quantum circuit is a general, well known, computational task. A device which qualitatively outperforms state-of-the-art classical computers in this task is clearly not simply a device 'simulating itself'.

The evaluation of effective error models for large scale universal quantum circuits is a difficult theoretical and experimental problem due to their complex nature. Therefore, existing proposals involve an expensive additional unitary transformation to the initial state [49] or are restricted to non-universal circuits [75]. Our proposal based on experimental measurements of the cross entropy, represents a novel way of characterizing and validating digital error models, and open quantum system theory in general. The method introduced here can also be applied to other systems, such as continuous chaotic Hamiltonian evolutions.

ACKNOWLEDGMENTS

We specially acknowledge Mikhail Smelyanskiy, from the Parallel Computing Lab, Intel Corporation, who performed the simulations of circuits with 6 × 6 and 7 × 6 qubits and wrote Appendix B. We would like to acknowledge Michael Bremner and Ashley Montanaro for multiple suggestions, specially regarding Sec. V. We would like to thank Scott Aaronson, Austin Fowler, Igor Markov, Masoud Mohseni and Eleanor Rieffel for discussions. The authors also thank Jeff Hammond, from the Parallel Computing Lab, Intel Corporation, for his useful insights into MPI run-time performance and scalability. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DEAC02-05CH11231.

Appendix A: Residual correlations after discrete errors

In this appendix we analyze numerically the residual correlations between the output of an ideal circuit and the output when a single X error (bit-flip) or Z error (phase-flip) is applied to one of the qubits. This residual correlation is responsible for the slight upper curvature seen in the red line in Fig. 3. It is also principally responsible for the small disparity between the cross entropy difference and the estimated fidelity seen in Fig. 4.

Figure. 10 shows the residual correlation for a single Z error (phase-flip) applied at different depths. We see that a bit-flip does not affect the output distribution if it is applied close to the end of the circuit. The reason is that we measure in the computational basis, which is insensitive to phase errors. Furthermore, the two-qubit CZ gates used in the circuit commute with Z errors.

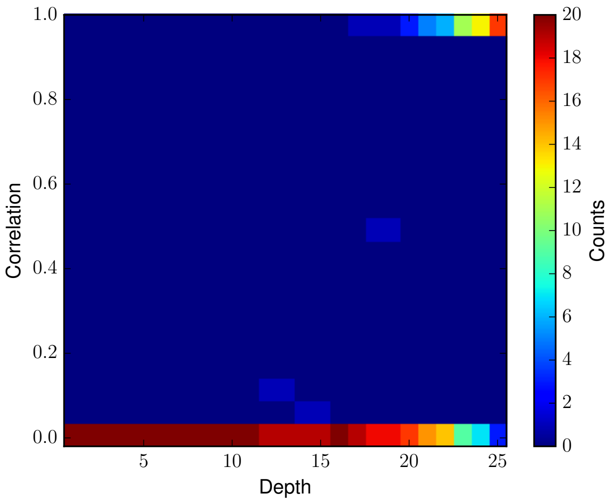

FIG. 10. Two-dimensional histogram of residual correlations after a single Z error (phase-flip) is applied at different depths. We calculate numerically the correlation between the output of the circuit of Fig. 3, with 5 × 4 qubits and total depth 25, and the output when a phase flip is applied to one of the 20 qubits.

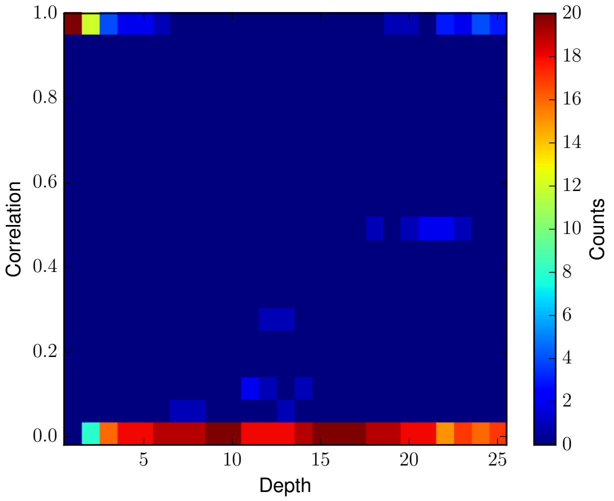

FIG. 11. Two-dimensional histogram of residual correlations for a single X error (bit-flip) applied at different depths. Same circuit as in Fig. 3 and Fig. 11.

Figure. 10 shows the residual correlation for a single X error (bit-flip). Bit-flip errors do not have any effect after the cycle of Hadamards at the beginning of the circuit (see Sec. IV), which rotate the initial state (in the computational basis) to the x basis. Some bit-flip errors towards the end of the circuit also do not affect correlations because the corresponding X error can get acted upon by a Hadamard-like gate, such as $\mathrm{Y}^{1/2}$. This rotates the X error into the z basis, in which the state is measured.

Appendix B: Quantum Simulation Details

In this appendix we summarize the implementation, optimization and performance of our high-performance gate-level quantum simulator. Additional details are available in [76, 77]. This simulation was used for all the circuits with 6×6 and 7×6 qubits. Simulations of smaller circuits, including all the simulations with errors, were performed with a different simulator running in local workstations.

In order to simulate quantum circuits on a classical computer, we implement a distributed high-performance quantum simulator that can simulate general single-qubit gates and two-qubit controlled gates. We perform a number of single- and multi-node optimizations, including vectorization, multi-threading, cache blocking, as well as gate specialization to avoid communication. Using Edison, distributed Cray XC30 system at National Energy Research Scientific Computing Center (NERSC), we simulate random quantum circuits of up to 42 qubits, with an average time per gate of 1.83 seconds. These are the largest quantum circuits simulated to-date for a computational task that approaches quantum supremacy.

1. Background

Given n qubits, our simulator evolves a 2^n state vector, using single-qubit as well as two-qubit controlled gates. Let U_{sq} be a 2×2 unitary matrix that represents a single-qubit gate operation:

$$U_{\mathrm{sq}} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} .$$

To perform gate U_{sq} on qubit k of the n-qubit quantum register, we apply U_{sq} to the pairs of amplitudes whose indices differ in the k-th bits of their binary index:

$$\begin{aligned} \alpha'_{*\ldots*0_k*\ldots*} &= u_{11} \cdot \alpha_{*\ldots*0_k*\ldots*} + u_{12} \cdot \alpha_{*\ldots*1_k*\ldots*} \\ \alpha'_{*\ldots*1_k*\ldots*} &= u_{21} \cdot \alpha_{*\ldots*0_k*\ldots*} + u_{22} \cdot \alpha_{*\ldots*1_k*\ldots*} \end{aligned} \qquad \text{(B1)}$$

A generalized two-qubit controlled-U gate, with a control qubit c and a target qubit t, works similarly to a single-qubit gate, except that only the pairs of amplitudes for which c is set are affected, while all other amplitudes are left unmodified.

2. Implementation and Optimization

The implementation of single- and two-qubit controlled gates follows directly Eq. B1. For example, to apply a single-qubit gate to qubit k, we iterate over consecutive groups of amplitudes of length 2^{k+1}, applying U_{sq} to every pair of amplitudes that are 2^k elements apart. To achieve high performance, we perform the following optimizations.

Vectorization: Exploring data parallelism is fundamental to the high performance and energy efficiency of modern architectures. Modern Intel CPUs support data parallelism in the form of SIMD (Single Instruction Multiple Data) instructions, such as AVX2 [78]. These instructions perform four double-precision operations simultaneously on four elements of the input registers. Our implementation maps every two pairs of complex amplitudes into four-wide SIMD instructions; each pair, which operates on real and imaginary parts, uses half of the SIMD register.[5]

Multithreading: Modern multi- and many-core CPUs support execution of many concurrent hardware threads. We parallelize single- and two-qubit controlled gate operations on these threads using OpenMP 4.0 [79]. We adaptively exploit thread-level parallelism either across groups or within a single group. Namely, we first try to divide groups of amplitudes evenly among all threads. When there are not enough groups to use all available threads, we explore thread parallelism within a group.

Cache Blocking: Single and controlled qubit operations perform a small amount of computation, and, as a result, their performance is limited by memory bandwidth. To increase arithmetic intensity of the quantum simulator, one can form larger gate matrices as a tensor product of several parallel gates. As a result, subsets of amplitudes are reused over matrix columns, but at the expense of redundant computation, which grows exponentially with the number of combined gates. Our approach identifies and operates on groups of consecutive gates which update a small portion of the state vector, common to all the gates, that also fits into Last Level Cache (LLC). LLC offers much higher bandwidth than main memory, which improves the performance of the simulator. LLC also has much smaller capacity, which limits this optimization only to the gates that operate on lower-order qubits [76].

Multi-node Implementation: Single node quantum simulation is limited by the size of the physical memory of the compute node.[6] To simulate larger numbers of qubits requires a distributed implementation. Our distributed simulation partitions a state vector of 2^n amplitudes (2^{n+4} bytes) among 2^p nodes, such that each node stores a local state of 2^{n-p} amplitudes. Given single- or controlled two-qubit gate operations on the target qubit

[5] Intel recently announced that the second generation Intel® Xeon Phi™ architecture will also support eight-wide AVX512. This will allow simultaneous operations on four pairs of amplitudes, and will enable additional performance benefits.

[6] While is conceivable to hold the state on the secondary storage device, the latter is significantly slower than main memory, thus rendering most interesting quantum simulations unpractical.

k, if $k < n - p$, the operation is fully contained within a node; otherwise it requires inter-node communication. Our communication scheme follows [80], where two nodes exchange half of their state vectors into each other's temporary storage, compute on exchanged halves, followed by another pair-wise exchange. In contrast to [80] which requires large temporary space to hold exchanged halves, our implementation requires very small temporary storage and is thus much more memory efficient.

Gate Specialization [77, 81]. To further reduce the run-time of the simulator, we take advantage of the specialized structure of each gate matrix. For example, the entries of a Hadamard matrix are real, which reduces the extra overhead of complex arithmetic. This is particularly helpful when combined with cache blocking which makes the simulation more compute bound. Recognizing diagonal gates, such as T gates, allows one to avoid inter-node communication, while recognizing an entry equal to 1.0 on the main diagonal of the diagonal gates (as in Z or T gates), reduces memory bandwidth requirements by 2×, and results in commensurate performance improvements.

3. Performance

We performed quantum simulations on Edison supercomputer [82]. Edison is a distributed Cray XC30 system at National Energy Research Scientific Computing Center (NERSC), ranks # 39 in the latest TOP500 list, and consists of 5,576 compute nodes. Each node is a dual-socket Intel®Xeon E5 2695-V2 processor with 12 cores per socket, each running at 2.4GHz. Each core is a superscalar, out-of-order core that supports 2-way hyperthreading and offers AVX support. All 12 cores share a 30MB L3 last level cache and a memory controller connected to four DDR3-1600 DIMMs that together provide 64GB of memory per node (32GB per socket). The nodes are connected via Cray Aries with Dragonfly topology. We use OpenMP 4.0 [79] to parallelize computation among threads. We also use Intel® Compiler v15.0.1 and Intel® Cray MPI 7.3.1 library.

The time to simulate an n-qubit quantum circuit on 2^p nodes is proportionate to

$$f \frac{G2^{n-p}}{B_{\text{mem}}} + (1-f)\left(\frac{G2^{n-p}}{B_{\text{mem}}} + \frac{G2^{n-p}}{B_{\text{net}}}\right) .$$

Here, G is the total number of gates, B_{mem} is achievable memory bandwidth, B_{net} is achievable bidirectional network bandwidth, and f is the fraction of gates which do not require communication. The first term gives the time to simulate gates that do not require communication, while the second term gives the time to simulate gates that communicate. Thus we expect gate operations which require communication to be $1 + B_{\text{mem}}/B_{\text{net}}$ slower than gates which communicate. On Edison, the highest achievable memory bandwidth is 50 GB/s per socket, while the highest achievable bidirectional network bandwidth is 7 GB/s per socket [83]. Thus the expected slowdown of gates that require communication, compared to gates that do not, is ∼ 8×.

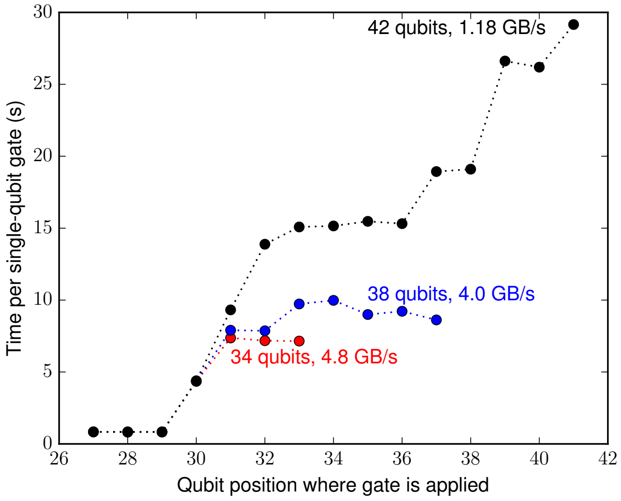

FIG. 12. Gate benchmarking results on multiple nodes (sockets) for the single-qubit Hadamard gate. The x-axis is the position of the qubit where the gate is applied. Operations on qubits in position 30 and above require network communication. The magnitude of the jump in the time per gate after position 30 is commensurate with the ratio between network and memory bandwidth. Numbers in the labels show achieved bandwidth for the higher ordered qubits.

Figure 12 reports benchmarks of the performance of a single-qubit Hadamard gate on 16, 256, and 4,096 sockets, simulating 34, 38, and 42 qubits, respectively, while keeping the problem size per socket constant (i.e., 2^{30} double complex amplitudes, or 2^{34} bytes). Gates performed on qubits $0 - 30$ require no inter-socket communication and take ∼0.82 seconds per gate. This corresponds to 42 GB/s memory bandwidth (2 [accesses (read/write)] $\cdot 2^{34}$ [bytes] / 0.82 [seconds]), or 84% of highest achievable bandwidth.

Gates applied to higher-order qubits, 30 and above, require communication, which increases the time per gate. For example, for a 36-qubit system simulated on 16 sockets, the time per gate increases to 7.6 seconds, which corresponds to 4.8 GB/s network bandwidth. The 9× increase compared to the no-communication case is consistent with our expectation, discussed earlier. As we increase the number of sockets, the time per gate further increases for higher order qubits. For example, for a 42-qubit system on 4,096 sockets, the time to apply a Hadamard gate to qubit 41 is 29 seconds – a nearly threefold increase compared to applying a Hadamard gate to qubit 31. This corresponds to 1.18 GB/s network bandwidth, which is almost a 6× drop, compared to the best achievable bandwidth of 7 GB/s. This drop is consistent with the detailed bandwidth analysis of Aries interconnect in Ref. [83]. Intuitively, the drop is due to the fact that higher-order qubits result in a larger distance between communicating sockets, which, in turn, results in increased volume of communication over global links and thus strains the bi-section bandwidth of the system.

Table I compares simulator performance characteristics of five random circuits with different lattice dimensions and number of qubits. The table is broken into five sections, one for each circuit. For each circuit, we show the characteristics for three levels of optimization: without specialization, with specialization, and with both specialization and cache blocking (*cb*) enabled. Circuits with 20, 24 and 30 qubits are simulated on a single socket, while circuits with 36 and 42 qubits are simulated on 64 and 4,096 sockets, respectively.

Specializing the gates reduces run-time of a 20-qubit circuit by 1.46×, compared to 1.26× run-time reduction for 24- and 30-qubit circuits, as shown in the first three sections of the table. As mentioned in Section B 2, specializing gates, such as T and CZ, reduces memory traffic by 2×. This reduces the simulation time of these gates on 24- and 30- qubit systems, whose state does not fit into Last Level Cache (LLC), making their performance bounded by memory bandwidth. In addition to 2× reduction in memory traffic, gate specialization also reduces compute requirements by as much as 4×: for example, without specialization, applying a T gate results in four complex multiply-adds per pair of state elements, while with specialization applying a T gate results in only one complex multiply-add. This reduces the simulation time of a 20-qubit system, whose 17 MB state fits into the 30 MB of the Last Level Cache (LLC), making its performance compute-bound. Thus gate specialization results in higher run-time reduction for a 20-qubit circuit than for 24- and 30-qubit circuits. Another consequence of the fact that the state of a 20-qubit circuit fits into LLC is that cache blocking optimization does not take effect. Furthermore, for 24- and 30-qubit circuits, cache blocking reduces the average time per gate by 2.1× and 1.6×, respectively. A 30-qubit circuit benefits less from cache blocking, compared to a 24-qubit circuit, because it has fewer gates that can be fused, as shown in the fifth column.

The last two sections of the table show performance statistics for a 36- and a 42-qubit circuits, which are simulated on 64 and 4,096 sockets, respectively. As Figure 12 shows, for a 36-qubit simulation the time per gate varies between 0.8 seconds (when there is no communication) and 8 seconds (when communication is required). Note that only 16% of the gates require communication, as shown in the second column. As a result, we measure an average time of 1.5 seconds per gate, as shown in the fourth column of the table. Gate specialization more than halves the number of gates that require communication. This results in 1.08 seconds per gate: 1.4× reduction of average time per gate, compared to no specialization. Combining cache blocking optimization with specialization reduces the time per gate down to 0.76 seconds: an additional 1.4× reduction compared to specialization only. As shown in the fourth column, for a 36-qubit circuit, we are able to fuse over five consecutive gates, on average. Overall, both gate specialization and cache blocking reduce the average time per gate as well as the total run-time of a circuit with depth 25 (last column) by nearly 2×.

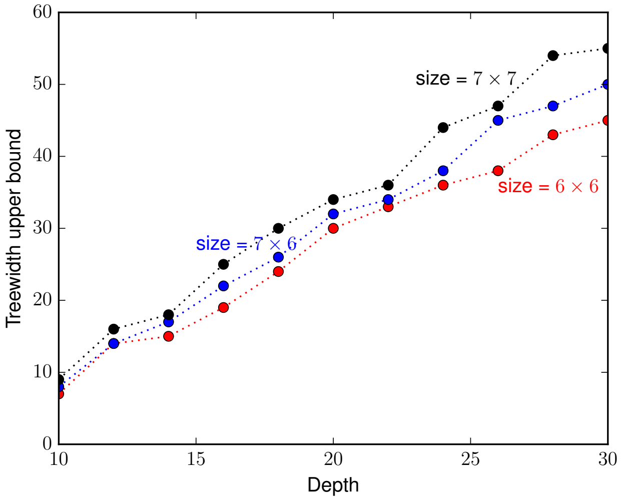

FIG. 13. Numerical upper bound for the treewidth of the tensor contraction of circuits with 6 × 6, 7 × 6, and 7 × 7 qubits as a function of the circuit depth.

The last row shows the simulator performance on a 42-qubit random circuit when both gate specialization and cache blocking are used. Compared to a 36-qubit random circuit, the number of gates per level on a 42-qubit random circuit has increased by almost 20%. In addition, as the second column shows, the fraction of gates that requires communication has increased by almost 2×, while the time per gate has also increased, as shown in Figure 12. As a result, the average time per gate on a 42-qubit simulation is 1.72 seconds; a 2.3× increase compared to a 36-qubit simulation. Overall, it took 1,589 seconds to simulate a 42-qubit circuit with the depth of 25: 989 seconds (1.72 seconds per gate × 23.0 gates per level × 25 levels) to simulate all the gates, and 600 seconds to compute statistics, such as entropy, the cross entropy with the uniform distribution and probability moments.

Appendix C: Numerical estimation of treewidth

The probability $p_U(x)$ of obtaining bit-string x as the output of a quantum circuit U can be obtained with an algorithm known as tensor network contraction [42]. The run time of this algorithm is exponential in the treewidth of a graph related to the layout of the gates in the circuit. For a circuit in a 2D lattice of qubits with two-qubit gates

Optimization Level	**% of comm**	**# of sockets**	**# of fused**	**Avg. time per gate (sec)**	**Time per Depth-25 (sec)**
5×4 *circuit*: 20 qubits, 10.3 gates per level, **17 MB of memory**					
no spec	0.0%	1	n/a	0.00022	0.057
spec	0.0%	1	n/a	0.00015	0.039
spec+cb	0.0%	1	0.00	0.00015	**0.039**
6×4 *circuit*: 24 qubits, 12.5 gates per level, **268 MB of memory**					
no spec	0.0%	1	n/a	0.0111	3.466
spec	0.0%	1	n/a	0.0088	2.741
spec+cb	0.0%	1	7.01	0.0041	**1.294**
6×5 *circuit*: 30 qubits, 16.2 gates per level, **17 GB of memory**					
no spec	0.0%	1	n/a	0.721	292.2
spec	0.0%	1	n/a	0.572	231.8
spec+cb	0.0%	1	5.64	0.349	**141.3**
6×6 *circuit*: 36 qubits, 19.5 gates per level, **1 TB of memory**					
no spec	15.9%	32	n/a	1.51	735.1
spec	6.2%	32	n/a	1.08	526.7
spec+cb	6.2%	64	5.40	0.76	**369.0**
7×6 *circuit*: 42 qubits, 23.0 gates per level, **70 TB of memory**					
spec+cb	11.2%	4,096	5.54	1.72	**989.0**

TABLE I. Simulator performance comparison of five random circuits: 5×4, 6×4, 6×5, 6×6, and 7×6 . First column lists three levels of optimizations, for each circuit. Second column shows the fraction of gates which require communication $(1 - f)$. Third and fourth columns show the number of sockets used, and average number of fused gates to enable cache blocking (*cb*) optimization, respectively (see Sec. B 2). The last two columns show average time per gate and time per circuit with depth 25, respectively.

restricted to nearest neighbors, the treewidth is proportional to $\min(d\sqrt{n}, n)$. Figure 13 shows numerical upper bounds for the treewidth as a function of depth for the circuits in Sec. IV. The upper bound was obtained by running the *QuickBB* algorithm [84]. We run this algorithm for each instance and depth for 100K seconds, more than 11.5 days. Typically the upper bound did not change after the first day. The treewidth of random circuits with 7×6 qubits becomes intractably large for tensor contractions when the gate depth exceeds approximately 25.

Appendix D: Non-Clifford gates

Clifford circuits (circuits which only contain Clifford gates) can be simulated efficiently [61]. Furthermore, this method can be extended to simulate circuits which are dominated by Clifford gates [44]. The only non-Clifford gate employed on the circuits we have used, as defined in Sec. IV, is the T gate. Figure 14 plots the number of T gates. On the one hand, the number of T gates is likely too big for this simulation method to work for circuits with 7×7 qubits and depth 25. This number can also be easily increased. On the other hand, the number of T gates can be decreased at will, which will allow for the verification of circuits with even 7×7 qubits, when a direct simulation is likely no longer possible.

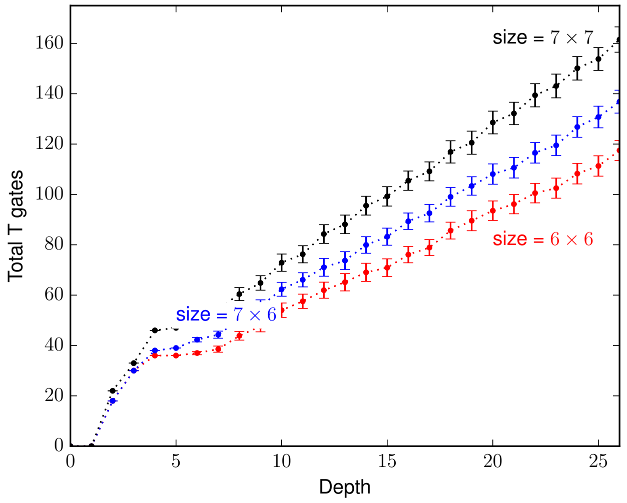

FIG. 14. Number of non-Clifford T gates as a function of depth for circuits with 6×6, 7×6, and 7×7 qubits. Error bars are the standard deviations among random circuit instances.

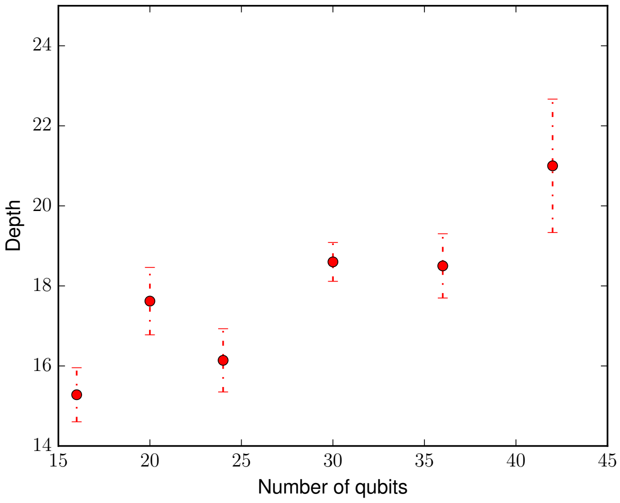

FIG. 15. First cycle in a random circuit instance such that the entropy remains within 4-sigma of the Porter-Thomas entropy during all the following cycles. Markers show the mean among instances and error bars correspond to the standard deviation among circuit instances. Depth is measured in layers, and each layer is a cycle of random single-qubit followed by a cycle of CZ gates.

Appendix E: Depth to reach Porter-Thomas for denser 2D circuits

It is currently not possible to perform two CZ gates simultaneously in two neighboring superconducting qubits [31, 32, 45, 48]. This restriction was used for the circuits of the main text, see Fig. 6. In this appendix we report simulations of circuits in a 2D lattice where, ignoring this particular restriction, a two-qubit gate is applied to every qubit in each cycle of CZ gates. In order to get a smoother scaling for circuits of different sizes, we use periodic boundary conditions for the layout of two-qubit gates. We find numerically a good convergence to the Porter-Thomas distribution for the following circuits.

We use the same single-qubit gates as in the main text, $\{\mathrm{X}^{1/2}, \mathrm{Y}^{1/2}, \mathrm{T}\}$. In addition we use two-qubit CZ gates. The circuits are:

1. Initialize in the state $|0\rangle^{\otimes n}$.
2. Apply a Hadamard gate to each qubit.
3. Apply a random circuit with a stack of depth d, where each layer has the following two clock cycles:
   (a) Apply a clock cycle of random single-qubit gates to all qubits.
   (b) Apply a clock cycle of two-qubit CZ gates.

We follow the same restrictions for the placement of single-qubit gates as in Sec. IV. For the cycle of two-qubit gates, we follow a similar sequence to the layouts of Fig. 6, but now every qubit participates in exactly one CZ gate. In addition, as mentioned above, we use periodic boundary conditions.

Figure 15 shows the first layer of each random circuit instance for which the entropy remains within 4-sigma of the Porter-Thomas entropy during all the following layers (similar to Fig. 9). Note that we now measure the depth in layers, and each layer consists of a cycle of single-qubit gates and a cycle of two-qubit gates. Physically, though, cycles of single-qubit gates are normally faster than cycles of two-qubit gates.

Appendix F: Outline of Stockmeyer Counting Theorem

In this section we outline the main ideas behind the Stockmeyer Counting Theorem [58, 85, 86]. As discussed in Sec. V A an NP-oracle is a computational complexity theory construct that determines if a given equation

$$f(z) = x \tag{F1}$$

has any solutions, see for example Eq. (20). The function f maps bit-strings to bit-strings and can be evaluated in polynomial time in the input size n. The Stockmeyer Counting Theorem states that an NP-oracle also suffices to determine, with high probability, an approximation $\tilde{q}(x)$ to the number of solutions $q(x)$ of Eq. (F1)

$$|\tilde{q}(x) - q(x)| < q(x)/\mathrm{poly}(n) \tag{F2}$$

where poly(n) denotes any chosen polynomial in n. The main ingredient is the use of so-called hash functions, described below, to estimate if there are at least 2^k solutions. The result then follows by trying different values of $k \le n$.

A hash function $h_{n,m}$ maps an n-bit-string to an m-bit-string with $n > m$. Let's consider the subset T_h of bit-strings which are mapped to 0 by h. Let $H_{n,m}$ be a sufficiently random family of hash functions (a pairwise independent family). Let S be a subset of n-bit-strings of size $|S|$ sufficiently larger than 2^m. Because a random $h_{n,m} \in H_{n,m}$ selects a random T_h, the size of the subset $S \cap T_h$ is concentrated around its expectation value $|S|/2^m$ [87].

Consider now the set $S \equiv \{z : f(z) = x\}$ of solutions z to Eq. (F1). We can use a random family of hash functions to construct an algorithm that with finite probability of success, 3/4 for example, can distinguish between $|S| > 2^k$ and $|S| \le 2^k$, where $k = m + 5$. This is done using a single NP-oracle call to check if there are a finite number of elements S mapped to 0, 48 for example, by a random hash function from $H_{n,m}$. The probability of success can be amplified to $1 - 1/(4\kappa)$ with κ invocations of the NP-oracle.

Appendix G: Multiplicative approximation to $|Z|^2$ from the Porter-Thomas distribution

We recall from the discussion in Sec. V B that each output probability of a random quantum circuit $p_U(x)$ is proportional to the partition function of a complex Ising model. In this appendix we review why approximate sampling with constant variational distance from the output of random circuits implies a probabilistic multiplicative error approximation to such partition functions with an NP-oracle [22, 27]. We follow the proof from Ref. [27], but use the Porter-Thomas distribution, instead of their anti-concentration bound.

Let $q(x)$ denote the output probability of a classical sampling algorithm for a bit-string x of our choice, and $\tilde{q}(x)$ an approximation obtained using the Stockmeyer Counting Theorem. From Eq. (F2) and the triangle inequality we obtain

$$\begin{aligned}|\tilde{q}(x) - p(x)| \leq (1 + 1/\mathrm{poly}(n))\, |q(x) - p(x)| \\ + p(x)/\mathrm{poly}(n)\ . \quad \text{(G1)}\end{aligned}$$

Let us suppose what we want to disprove: a classical sampling algorithm $A_{\mathrm{pcl}}(U)$ with probabilities $q(x)$ and polynomial computational time in n which achieves an ϵ approximation in the variational distance to the output of any given quantum random circuit

$$\sum_x |q(x) - p_U(x)| < \epsilon\ . \tag{G2}$$

We will show that then $p_U(x)$ can be approximated using Stockmeyer Counting Theorem, which is conjectured to be impossible.

From Markov's inequality we have, for any $0 < \delta < 1$,

$$\mathrm{Pr}_x \left(|q(x) - p_U(x)| \geq \frac{\epsilon}{2^n \delta} \right) \leq \delta \tag{G3}$$

where x is picked uniformly at random. Setting $\delta = 4\epsilon$ we obtain

$$\mathrm{Pr}_x \left(|q(x) - p_U(x)| \leq \frac{1}{2^{n+2}} \right) \geq 1 - 4\epsilon\ . \tag{G4}$$

Therefore, with probability $1 - 4\epsilon$, we have

$$|\tilde{q}(x) - p_U(x)| \leq \frac{1 + 1/\mathrm{poly}(n)}{2^{n+2}} + p(x)/\mathrm{poly}(n)\ . \tag{G5}$$

Set, for example, $\epsilon = (8e)^{-1} \approx 0.046$. If, as found numerically in Sec. IV, we assume that the output of U has Porter-Thomas distribution, then

$$\mathrm{Pr}\left(p_U(x) > 2^{-n}\right) = 1/e\ . \tag{G6}$$

Eqs. (G5) and (G6) imply that $\tilde{q}(x)$ approximates $p_U(x)$ up to a multiplicative error $1/4 + o(1)$ with probability at least $1/e - 4\epsilon = (2e)^{-1}$. A similar bound can be found using only the second moment of the Porter-Thomas distribution [27].

From Sec. V B we have that $p_U(x) = \lambda |Z|^2$ where λ is a positive known constant and Z is the partition function of a complex Ising model. Therefore, if $\tilde{q}(x)$ approximates $p_U(x)$ up to a multiplicative error $1/4 + o(1)$, then $\tilde{q}(x)/\lambda$ approximates $|Z|^2$ up to the same multiplicative error.

Appendix H: Bayesian estimation of output probabilities

In this appendix we study a polynomial classical algorithm for approximately sampling the output distribution of a circuit U. The sampling follows from on an approximation to the output probability $p_U(x)$ of a bit-string x. As has been discussed in Sec. V B, the amplitudes of the output state of a random quantum circuit U can be written in the form of a Feynman path integral where each path is encoded in the assignment of the vector s of Ising spins, and the phase associated with the path is given by the energy of an Ising model $H_x(s)$. The approximation algorithm considered here is a Bayesian estimation of the output probability of a given bit-string after randomly sampling a large number Feynman paths.

The output amplitudes of a random circuit are proportional to the partition function of a random Ising model $H_x(s)$ at complex temperature,

$$\Psi = \langle x | \psi_d \rangle = \frac{1}{\sqrt{L}} \sum_{k=0}^{K-1} M_k e^{i \frac{2\pi}{K} k} \tag{H1}$$

where the k's are different energies of the Ising model (mod K), $L = 2^G$, $M_k \sim 2^G$, and G is the number of two-sparse gates. The prefactor is $1/\sqrt{L}$ given the explicit choice of two-sparse gates, see Eq. (24).

We can always attempt to approximate the amplitude Ψ for circuits of any size by sampling a large number Q of spins configurations s in the partition function. We start by counting the number of configurations Q_k for each phase $k \in [0\ ..\ K-1]$ using the Ising model $H_x(s)$. We can assume that $1 \ll Q_k \ll L$. For example, the number of spins configurations is $L \sim 2^{250}$ for circuits with 7×6 qubits and depth 25. We will use the prior distribution from Porter-Thomas to derive the posterior distribution $\mathrm{Pr}(\Psi | \{Q_k\})$. We will see that the result is equivalent to a circuit fidelity $\sim Q^2/(NL)$. For instance, even if we sample $Q = 10^{18}$ spin configurations this will give a fidelity of approximately $\sim 10^{-52}$ for a circuit with 7×6 qubits and depth 25.

Define the probabilities of the different paths as $p_k = M_k/L$. The prior probability of an amplitude from the Porter-Thomas distribution is

$$\begin{aligned}\mathrm{Pr}(\Psi) &\propto \exp(-N \Psi \Psi^*) \qquad \text{(H2)} \\ &= \exp\left(-NL \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} p_{k_1} p_{k_2} \cos \frac{2\pi}{K}(k_1 - k_2) \right)\ .\end{aligned}$$

We want to write the probabilities p_k in a basis v^α that diagonalizes the kernel $\cos\frac{2\pi}{K}(k_1-k_2)$,

$$\sum_{j=0}^{K-1}\cos\left(\frac{2\pi}{K}(m-j)\right)v_j^\alpha=\lambda_\alpha v_m^\alpha \tag{H3}$$

for $\alpha\in[0\,..\,K-1]$. The components v_j^α of the eigenvectors of the kernel are

$$v_j^0=\frac{1}{\sqrt{K}} \tag{H4}$$

$$v_j^\alpha=\sqrt{\frac{2}{K}}\cos\left(\frac{2\pi}{K}\alpha j\right)\,,\quad \alpha\in[1\,..\,K/2-1] \tag{H5}$$

$$v_j^{K/2}=\frac{(-1)^j}{\sqrt{K}} \tag{H6}$$

and

$$v_j^\alpha=\sqrt{\frac{2}{K}}\sin\left(\frac{2\pi}{K}(K-\alpha)j\right)\,,\ \alpha\in[K/2+1\,..\,K-1]\,.$$

The eigenvalues are

$$\lambda_\alpha=\frac{K}{2}\left(\delta_{\alpha,1}+\delta_{\alpha,K-1}\right)\ . \tag{H7}$$

Let c'_j be the components of the vector of probabilities p_k in the basis v^α. We renormalize them to $c_j=c'_j$ for $j\notin\{1,K-1\}$ and $c_{\{1,K-1\}}=\sqrt{NLK/2}\,c'_{\{1,K-1\}}$ to write

$$\Pr(\Psi)\propto\exp(-(c_1^2+c_{K-1}^2)) \tag{H8}$$

and

$$\begin{aligned}p_j=\frac{2}{K}\sqrt{\frac{1}{LN}}\left(c_1\cos\left(\frac{2\pi}{K}j\right)+c_{K-1}\sin\left(\frac{2\pi}{K}j\right)\right)\\+\frac{1}{K}+\sum_{\alpha=2}^{K-2}c_\alpha v_j^\alpha\ .\end{aligned} \tag{H9}$$

We define

$$\rho_k\equiv\sum_{\alpha=2}^{K-2}c_\alpha v_k^\alpha\ . \tag{H10}$$

With this definition, the numbers ρ_k obey the following constraints

$$\begin{aligned}0=\sum_{k=0}^{K-1}\rho_k\cos\left(\frac{2\pi}{K}k\right)=\sum_{k=0}^{K-1}\rho_k\sin\left(\frac{2\pi}{K}k\right)\\=\sum_{k=0}^{K-1}\rho_k\ ,\end{aligned} \tag{H11}$$

which will be used later to simplify the posterior probability.

The posterior probability for Ψ is

$$\Pr(\Psi|\{Q_k\})\propto Q!\prod_{k=0}^{K-1}\frac{p_k^{Q_k}}{Q_k!}\Pr(\Psi) \tag{H12}$$

$$\propto\exp\left(\sum_{k=0}^{K-1}Q_k\log p_k\right)\exp\left(-\left(c_1^2+c_{K-1}^2\right)\right)\ .$$

The log posterior for c_1,c_{K-1} is

$$\begin{aligned}&\log\Pr\left(c_1,c_{K-1},\{\rho_j\}|\{Q_k\}\right)\\&\qquad\propto\sum_{j=0}^{K-1}Q_j\log\left(p_j\right)-\left(c_1^2+c_{K-1}^2\right)\ .\end{aligned} \tag{H13}$$

We are interested in the posterior probability $p=|\Psi|^2$. Note that $Np=c_1^2+c_{K-1}^2$, as seen in the Porter-Thomas form of Eq. (H8). Therefore

$$\begin{aligned}\Pr(p,\{\rho_j\}|\{Q\})=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Pr\left(c_1,c_{K-1},\rho|Q\right)\\\delta\left(\frac{c_1^2+c_{K-1}^2}{N}-p\right)dc_1dc_2\ .\end{aligned} \tag{H14}$$

After a change of variables $c_1=r\cos\phi$, $c_{K-1}=r\sin\phi$ we obtain

$$\begin{aligned}&\Pr(p,\{\rho_j\}|\{Q_j\})=\\&\frac{N}{2}\int_0^{2\pi}\Pr\left(\sqrt{Np}\cos\phi,\sqrt{Np}\sin\phi,\rho|\{Q\}\right)d\phi\ .\end{aligned} \tag{H15}$$

Using Eq. (H9) in Eq. (H13) we write the Taylor series for the log posterior for c_1,c_{K-1} as

$$\begin{aligned}&\log\Pr\left(\sqrt{Np}\cos\phi,\sqrt{Np}\sin\phi,\{\rho_j\}|\{Q_k\}\right)\propto\\&\sum_{q=1}^{\infty}\frac{(-1)^{q+1}}{qL^{q/2}}p^{q/2}\sum_{j=0}^{K-1}Q_j\left(\frac{2\cos\left(\frac{2\pi}{K}j-\phi\right)}{K\rho_j+1}\right)^q\\&\qquad+\sum_{j=0}^{K-1}Q_j\log\left(\rho_j+\frac{1}{K}\right)-Np\ .\end{aligned} \tag{H16}$$

We keep only the first term in Q/L (using $1\ll Q\ll L$), which is

$$\begin{aligned}&\log\Pr\left(\sqrt{Np}\cos\phi,\sqrt{Np}\sin\phi,\{\rho_j\}|\{Q_k\}\right)\propto\\&\sqrt{\frac{p}{L}}\sum_{j=0}^{K-1}Q_j\frac{2\cos\left(\frac{2\pi}{K}j-\phi\right)}{K\rho_j+1}\\&\qquad+\sum_{j=0}^{K-1}Q_j\log\left(\rho_j+\frac{1}{K}\right)-Np\ .\end{aligned} \tag{H17}$$

Exponentiating we get the posterior distribution

$$\Pr\left(\sqrt{Np}\cos\phi, \sqrt{Np}\sin\phi, \{\rho_j\}|\{Q_k\}\right) \propto \\ e^{-Np}\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\rho_j + \frac{1}{K}\right)\right) \\ \left(1 + \sqrt{\frac{p}{L}}\sum_{j=0}^{K-1} Q_j \frac{2\cos\left(\frac{2\pi}{K}j - \phi\right)}{K\rho_j + 1} \right. \\ \left. + \frac{2p}{L}\left(\sum_{j=0}^{K-1} Q_j \frac{\cos\left(\frac{2\pi}{K}j - \phi\right)}{K\rho_j + 1}\right)^2\right) . \quad \text{(H18)}$$

Note that we keep the second term when exponentiating, which is order Q^2/L, but we drop the second term in Eq. (H16), which is or order $Q/L^{3/2}$.

We can carry out a further simplification by noticing that ρ_j, which is defined in Eq. (H10) from the vector of probabilities p_j, obeys $\rho_j \ll Q$. Therefore, from the form of Eq. (H18), we see that $\Pr(p|\{Q_k\}) \simeq \Pr(p, \{\bar{\rho}_j\}|\{Q_k\})$, where $\bar{\rho}_j$ is the expectation value of ρ_j consistent with $\{Q_k\}$. This value can be obtained maximizing the posterior Eq. (H13) subject to the constraints given in Eq. (H11).

We now insert Eq. (H18) into Eq. (H15) and carry out the integration to obtain

$$\Pr(p|\{Q_k\}) = Ce^{-Np}\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\bar{\rho}_j + \frac{1}{K}\right)\right) \\ \left(1 + \frac{p}{L}\sum_{j_1,j_2=0}^{K-1} Q_{j_1}Q_{j_2}\frac{\cos\left(\frac{2\pi(j_1-j_2)}{K}\right)}{\left(K\bar{\rho}_{j_1}+1\right)\left(K\bar{\rho}_{j_2}+1\right)}\right) . \quad \text{(H19)}$$

Equation (H19) is the posterior probability $\Pr(p|\{Q_k\})$ for an approximation of the output probability $p = p_U(x)$ of a bit-string x after sampling a large number Q of spin configurations or Feynman paths in the expression for Ψ. We see that the probability p enters explicitly in the last term, which is of the order Q^2/L. Next, we interpret this equation more formally.

We argued in the text that the output state $\rho_{\mathcal{K}}$ of an implementation with fidelity α of a quantum circuit U can be modeled with Eq. (18)

$$\rho_{\mathcal{K}} = \alpha\left|\psi_d\right\rangle\left\langle\psi_d\right| + (1-\alpha)\frac{\mathbb{1}}{N} . \quad \text{(H20)}$$

Then, the probability $p_U(x)$ for bit-strings x sampled from an implementation with fidelity α has a distribution

$$\Pr_\alpha(p_U(x)) = N^2 e^{-Np}\left(\alpha p + \frac{1-\alpha}{N}\right) , \quad \text{(H21)}$$

see also Eq. (19). We can compare the posterior distribution, given by Eq. (H19), with Eq. (H21) to obtain an equivalent "fidelity" α for the Bayesian classical approximate sampling algorithm, $\Pr(p|\{Q_k\}) = \Pr_\alpha(p_U(x))$.

First we obtain an expression for the normalization constant C from the p-independent equation

$$C\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\bar{\rho}_j + \frac{1}{K}\right)\right) = (1-\alpha)N . \quad \text{(H22)}$$

The equation linear in p gives

$$N^2\alpha = C\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\bar{\rho}_j + \frac{1}{K}\right)\right) \\ \frac{1}{L}\sum_{j_1,j_2=0}^{K-1} Q_{j_1}Q_{j_2}\frac{\cos\left(\frac{2\pi(j_1-j_2)}{K}\right)}{\left(K\bar{\rho}_{j_1}+1\right)\left(K\bar{\rho}_{j_2}+1\right)} . \quad \text{(H23)}$$

Solving for α we obtain

$$\alpha = \frac{1}{NL}\sum_{j_1,j_2=0}^{K-1} Q_{j_1}Q_{j_2}\frac{\cos\left(\frac{2\pi(j_1-j_2)}{K}\right)}{\left(K\bar{\rho}_{j_1}+1\right)\left(K\bar{\rho}_{j_2}+1\right)} . \quad \text{(H24)}$$

This is the final result, which shows that the equivalent circuit fidelity of the approximate sampling algorithm is $\alpha \sim Q^2/NL$, as promised.

[1] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).

[2] P. W. Shor, FOCS **35**, 124 (1994).

[3] C. Porter and R. Thomas, Phys.Rev. **104**, 483 (1956).

[4] A. Peres, Phys. Rev. A **30**, 1610 (1984).

[5] R. Schack and C. M. Caves, Phys. Rev. Lett. **71**, 525 (1993).

[6] C. W. Beenakker, Rev. Mod. Phys. **69**, 731 (1997).

[7] J. Emerson, Y. S. Weinstein, M. Saraceno, S. Lloyd, and D. G. Cory, Science **302**, 2098 (2003).

[8] T. Gorin, T. Prosen, T. H. Seligman, and M. Žnidarič, Phys. Rep. **435**, 33 (2006).

[9] O. C. Dahlsten, R. Oliveira, and M. B. Plenio, J. Phys. A **40**, 8081 (2007).

[10] A. Ambainis and J. Emerson, in *CCC'07* (IEEE, 2007) pp. 129–140.

[11] L. Arnaud and D. Braun, Phys. Rev. A **78**, 062329 (2008).

[12] A. W. Harrow and R. A. Low, Comm. Math. Phys. **291**, 257 (2009).

[13] Y. S. Weinstein, W. G. Brown, and L. Viola, Phys. Rev. A **78** (2008).

[14] W. G. Brown and L. Viola, Phys. Rev. Lett. **104**, 250501 (2010).

[15] W. Brown and O. Fawzi, arXiv:1210.6644 (2012).

[16] H. Kim and D. A. Huse, Phys. Rev. Lett. **111**, 127205 (2013).

[17] P. Hosur, X.-L. Qi, D. A. Roberts, and B. Yoshida, arXiv:1511.04021 (2015).

[18] S. Aaronson, QIC **3**, 165 (2003).

[19] B. M. Terhal and D. P. DiVincenzo, QIC **4**, 134 (2004).

[20] S. Aaronson, in *Proc. Roy. Soc. London Ser. A*, Vol. 461 (2005) pp. 3473–3482.

[21] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Proc. Roy. Soc. London Ser. A **467**, 459 (2011).

[22] S. Aaronson and A. Arkhipov, in *STOC* (ACM, 2011) pp. 333–342.

[23] K. Fujii and T. Morimae, arXiv:1311.2128 (2013).

[24] S. Aaronson, TOCS **55**, 281 (2014).

[25] K. Fujii, H. Kobayashi, T. Morimae, H. Nishimura, S. Tamate, and S. Tani, arXiv:1409.6777 (2014).

[26] R. Jozsa and M. Van Den Nest, QIC **14**, 633 (2014).

[27] M. J. Bremner, A. Montanaro, and D. J. Shepherd, arXiv:1504.07999 (2015).

[28] E. Farhi and A. W. Harrow, arXiv:1602.07674 (2016).

[29] J. Preskill, *25th Solvay Conf.*, (2012).

[30] M. L. Mehta, *Random matrices*, Vol. 142 (Academic press, 2004).

[31] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, and others, Nature **508**, 500 (2014).

[32] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, and others, Nature **519**, 66 (2015).

[33] J. M. Renes, R. Blume-Kohout, A. J. Scott, and C. M. Caves, J. Math. Phys. **45**, 2171 (2004).

[34] F. G. S. L. Brandao, A. W. Harrow, and M. Horodecki, arXiv:1208.0692 (2012).

[35] S. Lloyd, arXiv:1307.0378 (2013).

[36] S. Popescu, A. J. Short, and A. Winter, Nat. Phys. **2**, 754 (2006).

[37] C. Gogolin, M. Kliesch, L. Aolita, and J. Eisert, arXiv:1306.3995 (2013).

[38] C. Ududec, N. Wiebe, and J. Emerson, Phys. Rev. Lett. **111**, 080403 (2013).

[39] S. Aaronson and A. Arkhipov, QIC **14**, 1383 (2014).

[40] M. Walschaers, J. Kuipers, J.-D. Urbina, K. Mayer, M. C. Tichy, K. Richter, and A. Buchleitner, New J. Phys. **18**, 032001 (2016).

[41] J. Emerson, E. Livine, and S. Lloyd, Phys. Rev. A **72**, 060302 (2005).

[42] I. L. Markov and Y. Shi, SICOMP **38**, 963 (2008).

[43] H. Meuer, E. Strohmaier, J. Dongarra, and H. Simon, (2015).

[44] S. Bravyi and D. Gosset, arXiv:1601.07601 (2016).

[45] R. Barends, L. Lamata, J. Kelly, L. Garca-lvarez, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.-C. Hoi, C. Neill, P. J. J. OMalley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, E. Solano, and J. M. Martinis, Nat. Comm. **6**, 7654 (2015).

[46] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. Blakestad, J. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. Wineland, Phys. Rev. A **77**, 012307 (2008).

[47] G. G. Carlo, G. Benenti, G. Casati, and C. Mejia-Monasterio, Phys. Rev. A **69**, 062317 (2004).

[48] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. Fowler, B. Campbell, Y. Chen, *et al.*, Nature **534**, 222 (2016).

[49] J. Emerson, R. Alicki, and K. Zyczkowski, J. Opt B **7**, S347 (2005).

[50] E. Magesan, J. M. Gambetta, and J. Emerson, Phys. Rev. Lett. **106** (2011).

[51] E. Magesan, J. M. Gambetta, and J. Emerson, Phys. Rev. A **85** (2012).

[52] S. Boixo and A. Monras, Phys. Rev. Lett. **100**, 100503 (2008).

[53] J. Emerson, Science **302**, 2098 (2003).

[54] R. Oliveira, O. C. O. Dahlsten, and M. B. Plenio, Phys. Rev. Lett. **98** (2007).

[55] R. Beals, S. Brierley, O. Gray, A. W. Harrow, S. Kutin, N. Linden, D. Shepherd, and M. Stather, in *Proc. Roy. Soc. London Ser. A*, Vol. 469 (2013) p. 20120686.

[56] W. G. Brown, L. F. Santos, D. J. Starling, and L. Viola, Phys. Rev. E **77**, 021106 (2008).

[57] A. De Luca and A. Scardicchio, EPL **101**, 37003 (2013).

[58] L. Stockmeyer, in *STOC* (ACM, 1983) pp. 118–126.

[59] L. A. Goldberg and H. Guo, arXiv:1409.5627 (2014).

[60] G. De las Cuevas, M. Van den Nest, M. Martin-Delgado, *et al.*, New J. Phys. **13**, 093021 (2011).

[61] D. Gottesman, arXiv:quant-ph/9807006 (1998).

[62] G. Kalai and G. Kindler, arXiv:1409.3093 (2014).

[63] A. Arkhipov, Phys. Rev. A **92**, 062326 (2015).

[64] A. Leverrier and R. García-Patrón, QIC **15**, 0489 (2015).

[65] G. Kalai and G. Kindler, arXiv:1409.3093 [quant-ph] (2014), arXiv: 1409.3093.

[66] A. Arkhipov, Physical Review A **92** (2015), 10.1103/PhysRevA.92.062326, arXiv: 1412.2516.

[67] K. Fujii and S. Tamate, arXiv preprint arXiv:1406.6932 (2014).

[68] S. Trotzky, Y.-A. Chen, A. Flesch, I. P. McCulloch, U. Schollwöck, J. Eisert, and I. Bloch, Nat. Phys. **8**, 325 (2012).

[69] T. Lanting, A. Przybysz, A. Smirnov, F. Spedalieri, M. Amin, A. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. Hilton, E. Hoskinson, M. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M. Thom, E. Tolkacheva, S. Uchaikin, A. Wilson, and G. Rose, Phys. Rev. X **4** (2014).

[70] S. Boixo, T. F. Rnnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, Nat. Phys. **10**, 218 (2014).

[71] S. Boixo, V. N. Smelyanskiy, A. Shabani, S. V. Isakov, M. Dykman, V. S. Denchev, M. H. Amin, A. Y. Smirnov, M. Mohseni, and H. Neven, Nat. Comm. **7** (2016).

[72] T. Albash, T. F. Rønnow, M. Troyer, and D. A. Lidar, EPJ ST **224**, 111 (2015).

[73] S. V. Isakov, G. Mazzola, V. N. Smelyanskiy, Z. Jiang, S. Boixo, H. Neven, and M. Troyer, arXiv:1510.08057 (2015).

[74] Z. Jiang, V. N. Smelyanskiy, S. V. Isakov, S. Boixo, G. Mazzola, M. Troyer, and H. Neven, arXiv:1603.01293 (2016).

[75] S. T. Flammia and Y.-K. Liu, Phys. Rev. Lett. **106**, 230501 (2011).

[76] M. Smelyanskiy, N. P. D. Sawaya, and A. Aspuru-Guzik, (2016), arXiv:1601.07195.

[77] T. Häner, D. S. Steiger, M. Smelyanskiy, and M. Troyer, (2016), arxiv:1604.06460.

[78] “New Instruction Descriptions Now Available,” Software.intel.com, retrieved: 2012-01-17.

[79] OpenMP Architecture Review Board, "OpenMP application program interface version 3.0," (2013).

[80] D. B. Trieu, *Large-scale simulations of error prone quantum computation devices*, Ph.D. thesis, University of Wuppertal (2010).

[81] T. Häner, D. S. Steiger, M. Smelyanskiy, and M. Troyer, Personal communication (2016).

[82] "Edison Cray XC30," www.nersc.gov/systems/edison-cray-xc30, accessed: 2016-04-29.

[83] B. Austin, M. Cordery, H. Wasserman, and N. Wright, Cray Inc., (2013).

[84] V. Gogate and R. Dechter, in *Proc CUAI* (2004) pp. 201–208.

[85] L. Trevisan, "Lecture Notes on Computational Complexity," (2004).

[86] O. Goldreich, ACM SIGACT News **39**, 35 (2008).

[87] R. Impagliazzo, L. A. Levin, and M. Luby, STOC , 12 (1989).