nature physics

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41567-018-0124-x

In the format provided by the authors and unedited.

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], Michael J. Bremner[5], John M. Martinis[6,7] and Hartmut Neven[1]

[1]Google Inc., Venice, CA, USA. [2]Google Inc., Zurich, Switzerland. [3]QuAIL, NASA Ames Research Center, Moffett Field, CA, USA. [4]SGT Inc., Greenbelt, MD, USA. [5]Centre for Quantum Computation and Communication Technology, Centre for Quantum Software and Information, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, New South Wales, Australia. [6]Google Inc., Santa Barbara, CA, USA. [7]Department of Physics, University of California, Santa Barbara, CA, USA. *e-mail: boixo@google.com

Supplementary material for "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] Michael J. Bremner,[5] John M. Martinis,[6,7] 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, USA*
[5]*Centre for Quantum Computation and Communications Technology, Centre for Quantum Software and Information, University of Technology Sydney, NSW 2007, Australia*
[6]*Google Inc., Santa Barbara, CA 93117, USA*
[7]*Department of Physics, University of California, Santa Barbara, CA 93106, USA*

I. Porter-Thomas distribution

For completeness, we give here a direct derivation of the form of the distribution of the probability $p = |\langle x_0|\psi\rangle|^2$ of outcome $|x_0\rangle$ from a state $|\psi\rangle$ chosen uniformly at random in Hilbert space, over a complete basis $\{|x\rangle\}$ (see also Ref. [1]). We will show that this distribution is $\Pr(p) = (N-1)(1-p)^{N-2}$ where $N = 2^n$ is the size of Hilbert space (a Beta distribution). For large $N \gg 1$, this distribution is approximated well by $\Pr(p) = Ne^{-Np}$, an exponential distribution, which in this context we call the Porter-Thomas distribution [2–4].

We write a generic state in the basis $\{|x\rangle\}$ as $|\psi\rangle = \sum_x (a_x + ib_x)|x\rangle$. Assuming that $|\psi\rangle$ is chosen uniformly at random in Hilbert space the only constraint is normalization and we can write for $\Pr(p)$

$$\frac{\int_{-\infty}^{\infty} \Pi_x da_x db_x\, \delta(\sum_x a_x^2 + b_x^2 - 1)\delta(a_{x_0}^2 + b_{x_0}^2 - p)}{\int_{-\infty}^{\infty} \Pi_x da_x db_x\, \delta(\sum_x a_x^2 + b_x^2 - 1)} .$$

We calculate the numerator as

$$\begin{aligned}
&\int_{-\infty}^{\infty} \Pi_x da_x db_x\, \delta(\sum_x a_x^2 + b_x^2 - 1)\delta(a_{x_0}^2 b_{x_0}^2 - p) \\
&\quad = \int_{-\infty}^{\infty} \Pi_x da_x db_x \frac{1}{2\pi}\int_{-\infty}^{\infty} dt\, e^{it\sum_x (a_x^2+b_x^2)-it} \\
&\qquad \frac{1}{2\pi}\int_{-\infty}^{\infty} dw\, e^{iw(a_{x_0}^2+b_{x_0}^2)-iwp} \\
&\quad = \frac{1}{2\pi}\int_{-\infty}^{\infty} dt\, e^{-it}\left(\int_{-\infty}^{\infty} da_x db_x\, e^{it(a_x^2+b_x^2)}\right)^{N-1} \\
&\qquad \frac{1}{2\pi}\int_{-\infty}^{\infty} dw\, e^{-iwp}\int_{-\infty}^{\infty} da_x db_x e^{i(t+w)(a_x^2+b_x^2)} \\
&\quad = \frac{1}{2\pi}\int_{-\infty+i\epsilon}^{\infty+i\epsilon} dt\, e^{-it}\left(\frac{\pi}{-it}\right)^{N-1} \\
&\qquad \frac{1}{2\pi}\int_{-\infty+i\varepsilon}^{\infty+i\varepsilon} dw\, e^{-iwp}\frac{\pi}{-i(t+w)} \\
&\quad = \frac{\pi^N}{(-i)^N}\frac{1}{2\pi i}\int_{-\infty+i\epsilon}^{\infty+i\epsilon} dt\, e^{-it(1-p)}\frac{1}{t^{N-1}} \\
&\quad = -\frac{\pi^N}{(-i)^N}\frac{1}{(N-2)!}\lim_{t\to 0}\frac{d^{N-2}}{dt^{N-2}}e^{-it(1-p)} \\
&\quad = \pi^N \frac{1}{(N-2)!}(1-p)^{N-2} . \qquad (1)
\end{aligned}$$

The denominator is calculated in a similar way

$$\begin{aligned}
&\int_{-\infty}^{\infty} \Pi_x da_x db_x\, \delta(\sum_x a_x^2 + b_x^2 - 1) \\
&\quad = \frac{1}{2\pi}\int_{-\infty}^{\infty} dt\, e^{-it}\left(\int_{-\infty}^{\infty} da_x db_x\, e^{it(a_x^2+b_x^2)}\right)^N \\
&\quad = \pi^N \frac{1}{(N-1)!} \qquad (2)
\end{aligned}$$

Therefore

$$\Pr(p) = (N-1)(1-p)^{N-2} = N(e^{-Np} + O(1)) , \qquad (3)$$

where $O(1)$ denotes a correction independent of N. It can be seen that the correlation between the probability of different outputs $|x_j\rangle$ and $|x_k\rangle$ for fixed $|\psi\rangle$ is exponentially small in n [5, 6].

Equation (3) allows us to calculate other quantities of interest. We are particularly interested in the cross entropy obtained through measurements of an output state $\rho_{\mathcal{K}} = \alpha|\psi_d\rangle\langle\psi_d| + (1-\alpha)\frac{\mathbb{1}}{N}$ with fidelity α (see main text). The cross entropy is estimated sampling m bit-strings $\{x_j^{\rm exp}\}$, calculating the ideal probabilities $-\log p_U(x_j^{\rm exp})$, and averaging the corresponding random variable

$$\begin{aligned}
&-\frac{1}{m}\sum_j \log p_U(x_j^{\rm exp}) \to \\
&\qquad -\sum_x \left(\alpha\, p_U(x) + \frac{1-\alpha}{N}\right)\log p_U(x) . \qquad (4)
\end{aligned}$$

We calculate sums $\sum_x f(p_U(x))$ for a given function f using the expression $\frac{1}{N}\sum_x \delta(P_U(x) - p) = \Pr(p) = Ne^{-Np}$.

We obtain

$$\sum_x f(p_U(x)) = \sum_x f(p_U(x)) \int dp\, \delta(p_U(x) - p)$$
$$= \int dp \sum_x \delta(p_U(x) - p) f(p)$$
$$= N \int dp\, N e^{-Np} f(p) = N \langle f(p) \rangle \ . \quad (5)$$

Therefore

$$-\sum_x \left(\alpha\, p_U(x) + \frac{1-\alpha}{N} \right) \log p_U(x)$$
$$= -N \int dp\, N e^{-Np} \left(\alpha\, p + \frac{1-\alpha}{N} \right) \log p$$
$$= \log N - \alpha + \gamma \quad (6)$$

where γ is Euler's constant. Similarly we can calculate the variance of the random variable -$\log p_U(x)$ sampled from the output state $\rho_{\mathcal{K}}$, which is

$$\kappa^2 = \sum_x \left(\alpha\, p_U(x) + \frac{1-\alpha}{N} \right) \log^2 p_U(x)$$
$$- (\log N - \alpha + \gamma)^2$$
$$= \pi^2/6 - \alpha^2 \ . \quad (7)$$

Note that the variance κ^2 is independent of n.

II. Cross entropy benchmarking

We write the output of an experimental quantum circuit implementation with fidelity α_f as

$$\rho = \alpha_f U |\psi_0\rangle\langle\psi_0| U^\dagger + (1-\alpha_f)\sigma_U \ . \quad (8)$$

The state $U|\psi_0\rangle$ corresponds to the ideal output of circuit U, without any errors, while σ_U represents the output of the implementation of U after any experimental errors. The experimental output probability for a bit-string x is $p_{\rm exp}(x) = \langle x|\, \rho\, |x\rangle = \alpha_f\, p_U(x) + (1-\alpha_f)\, \langle x|\, \sigma_U\, |x\rangle$. From the arguments presented in the main text, and as seen in Fig 1a, almost any error in a universal random circuit of sufficient depth results in an output probability which is uncorrelated with the ideal probability distribution. Therefore we assume that $p_\sigma(x) = \langle x|\, \sigma_U\, |x\rangle$ is uncorrelated with $p_U(x)$.

The cross-entropy between a probability distribution p_A and the ideal output probability p_U is defined as $\mathrm{H}(p_A, p_U) = -\sum_x p_A(x) \log p_U(x)$. We average the cross entropy $\mathrm{H}(p_\sigma, p_U)$ over an ensemble $\{U\}$ of random circuits

$$\mathbb{E}_U \left[\mathrm{H}(p_\sigma, p_U) \right] = \mathbb{E}_U \left[\sum_{j=1}^N p_\sigma(x_j) \log \frac{1}{p_U(x_j)} \right] . \quad (9)$$

From the discussion above, we assume that averaging over the ensemble $\{U\}$ can be done independently, and we have

$$\mathbb{E}_U \left[\mathrm{H}(p_\sigma, p_U) \right] \approx -\sum_{j=1}^N \mathbb{E}_U \left[p_\sigma(x_j) \right] \mathbb{E}_U \left[\log p_U(x_j) \right] \quad (10)$$
$$= -\sum_{j=1}^N \mathbb{E}_U \left[\log p_U(x_j) \right] / N \quad (11)$$
$$= \mathbb{E}_U \left[-\sum_{j=1}^N \log p_U(x_j)/N \right] , \quad (12)$$

where we assume $\mathbb{E}_U \left[p_\sigma(x_j) \right] = 1/N$. We call this quantity, the average cross-entropy with the uniform distribution over bit-strings, $\mathrm{H}_{\rm unc}$.

We now look at the cross entropy difference. We have

$$\mathrm{H}_{\rm unc} - \mathbb{E}_U \left[\mathrm{H}(p_{\rm exp}, p_U) \right] \quad (13)$$
$$= \mathrm{H}_{\rm unc} + \mathbb{E}_U \Big[\sum_j \big(\alpha_f p_U(x_j)$$
$$+ (1-\alpha_f) \langle x_j|\, \sigma_U\, |x_j\rangle \big) \log p_U(x_j) \Big] \quad (14)$$
$$= \mathrm{H}_{\rm unc} - \alpha_f \mathbb{E}_U[\mathrm{H}(p_U)] - (1-\alpha_f)\mathrm{H}_{\rm unc} \quad (15)$$
$$= \alpha_f (\mathrm{H}_{\rm unc} - \mathbb{E}_U[\mathrm{H}(p_U)]) \ . \quad (16)$$

Therefore

$$\alpha_f = \frac{\mathrm{H}_{\rm unc} - \mathbb{E}_U[\mathrm{H}(p_{\rm exp}, p_U)]}{\mathrm{H}_{\rm unc} - \mathbb{E}_U[\mathrm{H}(p_U)]} \quad (17)$$

The cross-entropy $\mathrm{H}(p_{\rm exp}, p_U)$ can be estimated as explained in the main text using m bit-string measurements

$$\mathrm{H}(p_{\rm exp}, p_U) = -\frac{1}{m} \sum_{j=1}^m \log p_U(x_j^{\rm exp}) + O(1/\sqrt{m}) \ . \quad (18)$$

The other quantities can be estimated numerically (see also [7]).

Using the Porter-Thomas distribution, this method is simplified. 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, if we use random circuits of sufficient depth, we have (see previous section)

$$-\mathbb{E}_U \left[\log p_U(x_j) \right] \approx \log N + \gamma \ . \quad (19)$$

This implies that

$$\mathbb{E}_U \left[\mathrm{H}(p_\sigma, p_U) \right] \approx \log N + \gamma \ , \quad (20)$$

without assuming $\mathbb{E}_U \left[p_\sigma(x_j) \right] = 1/N$. We also have $E_U[\mathrm{H}(p_U)] = \log N - 1 + \gamma$. Therefore, Eq. (17) simplifies to

$$\alpha_f = \log N + \gamma - \mathbb{E}_U[\mathrm{H}(p_{\rm exp}, p_U)] \ . \quad (21)$$

III. 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 numerical simulations and 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 $p_U(x)$ [8–17]. In this section we will extend the method used to show the computational hardness of sampling commuting random circuits (IQP) [11, 16] to the general case of universal random circuits.

We will first describe the computational complexity of the problem of estimating a probability $p_{\rm pcl}(x)$ for any given 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

$$f(w, y) = x \ . \tag{22}$$

Here the bit-string $w = \{w_1, \ldots, w_k\}$ encodes the problem instance, y is a vector of random bits $y = \{y_1, \ldots, y_\ell\}$ chosen uniformly and x is the output bit-string. For fixed w and x, the number W_x of solution vectors y of Eq. (22) 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. (22) with fixed w and x belongs to the complexity class NP. The reason is that the polynomial time evaluation of f for given w and x can be translated into a polynomial size SAT formula with variables y by standard reductions, as in Cook's theorem [18]. 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 [19], states that probabilistically approximating the number of solutions W_x, and therefore $q(x)$, to within a multiplicative factor ϵ, can also be performed with an NP oracle in polynomial time in the input size and $1/\epsilon$, see Sec. III C and Ref. [20].

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 [11, 16]

$$\langle x|\psi\rangle = \lambda \sum_s e^{i\theta H_x(s)} \ , \quad H_x(s) = h_x \cdot s + s \cdot \hat{J} \cdot s \ , \tag{23}$$

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. (23) 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 NP oracle [11, 21, 22]. Reference [16] 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.

We argue by contradiction that is not possible to approximately sample from the distribution p_U classically efficiently. Assume 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. [16] that a fraction of these probabilities could be probabilistically approximated with multiplicative error using an NP-oracle, see Sec. IV. 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, and therefore can not be probabilistically approximated with multiplicative error using an NP-oracle, 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 the main text. Known algorithms for mapping universal quantum circuits to partition functions of complex Ising models use polynomial reductions to a universal gate set [23–25]. 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,

$$\langle x|\psi_d\rangle = \sum_{\{\sigma^t\}} \prod_{t=0}^{d} \langle \sigma^t| \, U^{(t)} \, |\sigma^{t-1}\rangle \, , \quad |\sigma^d\rangle = |x\rangle \, . \tag{24}$$

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 (24) 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,

$$s_j^{d(j)} = x^{(j)} \, , \quad j \in [1 \, .. \, n] \, . \tag{25}$$

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 (24) in the following form

$$\langle x|\psi_d\rangle = 2^{-G/2} \sum_s \exp\left(\frac{i\pi}{4} H_s(x)\right) \, . \tag{26}$$

Here $\exp(i\pi H_s(x)/4)$ is a phase factor associated with each path that depends explicitly on the end-point condition (25).

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

$$\frac{i\pi}{4} H_s^{\mathrm{X}^{1/2}}(x) = \frac{i\pi}{2} \sum_{j=1}^n \sum_{k=0}^{d(j)} \alpha_j^k \frac{1+s_j^{k-1}s_j^k}{2} \, , \tag{27}$$

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

As mentioned above, the dependence on x arises due to the boundary condition (25). 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

$$\frac{i\pi}{4} H_s^T(x) = \frac{i\pi}{4} \sum_{j=1}^n \sum_{t=0}^{d} \tau_j^t \frac{1-s_j^{d(j,t)}}{2} \, . \tag{28}$$

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

$$\frac{i\pi}{4} H_s^{\mathrm{CZ}}(x) = i\pi \sum_{i=1}^n \sum_{j=1}^{i-1} \sum_{t=0}^{d} z_{ij}^t \frac{1-s_i^{d(i,t)}}{2} \frac{1-s_j^{d(j,t)}}{2} . \tag{29}$$

One can see from comparing (26) with (27)-(29) 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

$$H_s(x) = H_s^{(0)} + H_s^{(1)} + H^{(2)} \,. \tag{30}$$

Here

$$H_s^{(0)} = \sum_{i=1}^{n} \sum_{k=1}^{d(i)-1} h_i s_i + \sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{k=1}^{d(i)-1} \sum_{l=1}^{d(j)-1} \mathcal{J}_{ij}^{kl} s_i^k s_j^l. \tag{31}$$

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

$$H_s^{(1)}(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{d(i)-1} b_{ij}^k s_i^k x^{(j)} \,. \tag{32}$$

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

$$h_i^k = \alpha_i^{k+1} - \alpha_i^k - \frac{1}{2} J_i^k - \sum_{j=1}^{n} \sum_{l=1}^{d(j)} J_{ij}^{k\,l} \tag{33}$$

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

$$\mathcal{J}_{ij}^{kl} = J_{ij}^{kl} + \frac{1}{2} \delta_{i,j} (\delta_{k-1,l} + \delta_{k,l-1}) \left(2\alpha_i^{(k+l+1)/2} - 1 \right) \tag{34}$$

where

$$J_{ij}^{kl} = \sum_{t=1}^{d} \delta_{k,d(i,t)} \delta_{l,d(j,t)} z_{ij}^t \,, \tag{35}$$

and

$$J_i^k = \sum_{t=1}^{d} \delta_{k,d(i,t)} \tau_i^t \,. \tag{36}$$

The coupling coefficients b_{ij}^k in (32) equal

$$b_{ij}^k = \delta_{k,d(i)-1} \delta_{ij} (2\alpha_j^{d(j)} - 1) + J_{ij}^{k d(j)} \,. \tag{37}$$

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 (31).

To understand the structure of the graph defined by the Ising couplings (34) 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 Sec. VI). 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 $\mathrm{X}^{1/2}$, $\mathrm{Y}^{1/2}$, 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. (34). 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

$$\Pr[J_{ij}^{kl} = r] \equiv P(r) = \sum_{q=0}^{\infty} p(r|q) p(q) \,, \tag{38}$$

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 $p(r|q) = \binom{q+1}{r} p_{\mathrm{CZ}}^r (1 - p_{\mathrm{CZ}})^{q+1-r}$ is the probability of having r CZ gates over $q+1$ layers applied between a given pair of neighboring qubits. Finally, we have for $P(r)$

$$P(r) = \begin{cases} \dfrac{1 - p_{\mathrm{CZ}}}{1 + p_{\mathrm{CZ}}/8}, & r = 0 \\ \dfrac{9}{1 + p_{\mathrm{CZ}}/8} \left(\dfrac{p_{\mathrm{CZ}}/8}{1 + p_{\mathrm{CZ}}/8} \right)^r & r > 0 \,. \end{cases} \tag{39}$$

For a square grid of qubits $p_{\mathrm{CZ}} \simeq 1/4$. One can see from (39) that for $r \geq 1$ the distribution $\Pr[J_{ij}^{kl} = r]$ decays exponentially with r and $P(r+1)/P(r) \simeq p_{\mathrm{CZ}}/8 \simeq 1/32$. Therefore, the most likely values of J_{ij}^{kl} are 0, corresponding to the probability $P(0) \simeq 1 - p_{\mathrm{CZ}}$, and 1, corresponding to the probability $P(1) \simeq 9 p_{\mathrm{CZ}}/8$. The high probability of having no traversal couplings between qubits relates to the comparatively slow growth of the circuit graph treewidth, see Sec. VII.

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 worldline. Not too close to the end of the circuit ($d - t \gg \sqrt{d}$) we have

$$\mathfrak{p}_k(t) = \binom{t-1}{k-1} \left(\frac{1}{3}\right)^{t-k} \left(\frac{2}{3}\right)^k, \quad \sum_{t=k}^{\infty} \mathfrak{p}_k(t) = 1, \tag{40}$$

Similarly, the probability $\mathfrak{p}^t(l)$ of having exactly l vertices lo-

cated within t layers of a given worldline ($t_j^l \leq t$) equals

$$\mathfrak{p}^t(l) = \binom{t}{l} \left(\frac{1}{3}\right)^{t-l} \left(\frac{2}{3}\right)^{l}, \quad \sum_{l=0}^{t} \mathfrak{p}^t(l) = 1 \; . \tag{41}$$

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

$$\mathfrak{p}(l|k) = \sum_t \mathfrak{p}^t(l)\mathfrak{p}_k(t) \; . \tag{42}$$

Approximating the binomial coefficients with the Stirling formula we obtain

$$\mathfrak{p}(l|k) \simeq \sqrt{\frac{3}{2\pi(k+l)}} \exp\left(-\frac{3(k-l)^2}{2(k+l)}\right) \; . \tag{43}$$

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 worldline corresponding to qubits located on a 2D lattice. According to (34), 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 worldlines. The probability distribution of the coupling coefficients has exponential form, Eq. (39). Differences between the vertex indices that are involved in the lateral couplings obey a local Gaussian distribution, Eq. (43).

Finally, note that Eq. (26) 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 Sec. VIII).

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 [21, 22, 26].

C. Outline of Stockmeyer Counting Theorem

In this section we outline the main ideas behind the Stockmeyer Counting Theorem [19, 20, 27]. As discussed in Sec. III A an NP-oracle is a computational complexity theory construct that determines if a given equation

$$f(z) = x \tag{44}$$

has any solutions, see for example Eq. (22). 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. (44)

$$|\tilde{q}(x) - q(x)| < q(x)/\mathrm{poly}(n) \tag{45}$$

where $\mathrm{poly}(n)$ 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 $k \leq n$.

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 $|S|/2^m$ [28].

Consider now the set $S \equiv \{z : f(z) = x\}$ of solutions z to Eq. (44). 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 $|S| \leq 2^k$, 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.

IV. Multiplicative approximation to $|Z|^2$ from the Porter-Thomas distribution

We recall from the discussion in Sec. III 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 [12, 16]. We follow the proof from Ref. [16], 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. (45) and the triangle inequality we obtain

$$|\tilde{q}(x) - p(x)| \leq (1 + 1/\mathrm{poly}(n))\, |q(x) - p(x)| + p(x)/\mathrm{poly}(n) \; . \tag{46}$$

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

$$\sum_x |q(x) - p_U(x)| < \epsilon \; . \tag{47}$$

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 $0 < \delta < 1$,

$$\Pr_x \left(|q(x) - p_U(x)| \geq \frac{\epsilon}{2^n \delta} \right) \leq \delta \tag{48}$$

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

$$\Pr_x \left(|q(x) - p_U(x)| \leq \frac{1}{2^{n+2}} \right) \geq 1 - 4\epsilon \, . \tag{49}$$

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

$$|\tilde{q}(x) - p_U(x)| \leq \frac{1 + 1/\mathrm{poly}(n)}{2^{n+2}} + p_U(x)/\mathrm{poly}(n) \, . \tag{50}$$

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

$$\Pr \left(p_U(x) > 2^{-n} \right) = 1/e \, . \tag{51}$$

Eqs. (50) and (51) 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 [16]. Specifically, the bound on the expectation value of approximate 2-designs from Ref. [29] Lemma 9 can be inserted in the Paley-Zygmund inequality of Ref. [16] to obtain an inequality similar to Eq. (51) (see also Ref. [30]).

From Sec. III 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.

V. Depth to reach Porter-Thomas

We also studied the expected convergence to Porter-Thomas with depth proportional to $\sqrt{n}$ using a stronger criterion than those presented in the main text. 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. **S1** 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 Sec. VI.

We note that a sublinear convergence to the low order moments of the Porter-Thomas distribution is faster than rigorously proven bounds for random circuits [29, 31]. The latest rigorous bound from Ref. [31] scales like $O(kn^{3/2})$ for 2D circuits, and would translate in a depth in the order of hundreds cycles for the second moment, and thousands for the 10th moment. As shown numerically in the main text, a very modest depth of 20 cycles is sufficient for up to the 10th moment, and low order moments seem to converge at approximately similar depths.

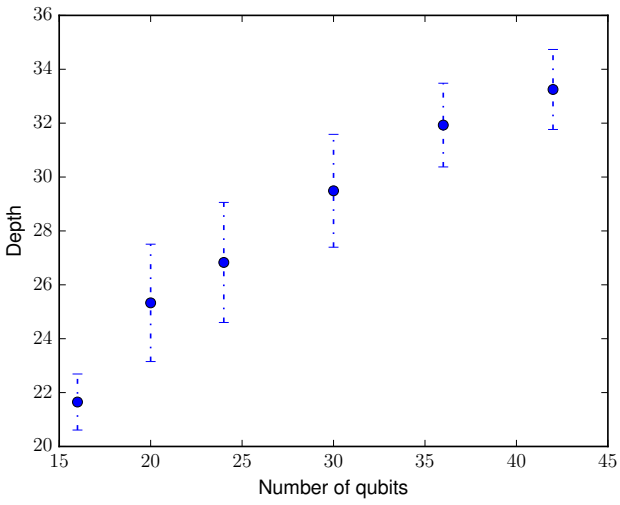

Fig. S1: First cycle in a random circuit instance such that the entropy remains within 4-sigma of the Porter-Thomas entropy during all the following cycles. Markers show the mean among instances and error bars correspond to the standard deviation among circuit instances.

Interestingly, sparse commuting random circuits (IQP) achieve a similar property (so-called anticoncentration) with depth proportional to $\sqrt{n}$, up to polylogarithmic factors in n, in a 2D lattice [32]. We have numerically verified that the output distribution of these circuits has the same entropy (up to small statistical fluctuations of order $2^{-n/2}$) as the Porter-Thomas distribution. The main reason to use random quantum circuits, instead of IQP circuits, is that IQP circuits are easier to simulate and harder to implement experimentally, see Sec. IX.

VI. 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 [33–36]. This restriction was used for the circuits of the main text. 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:

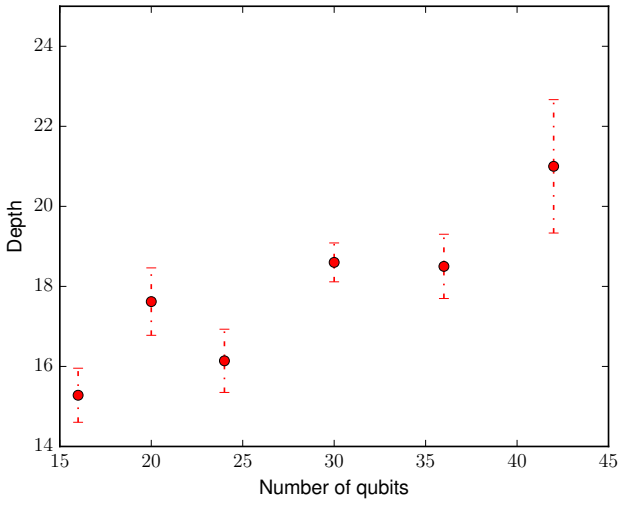

Fig. S2: First cycle in a random circuit instance such that the entropy remains within 4-sigma of the Porter-Thomas entropy during all the following cycles. Markers show the mean among instances and error bars correspond to the standard deviation among circuit instances. Depth is measured in layers, and each layer is a cycle of random single-qubit followed by a cycle of CZ gates.

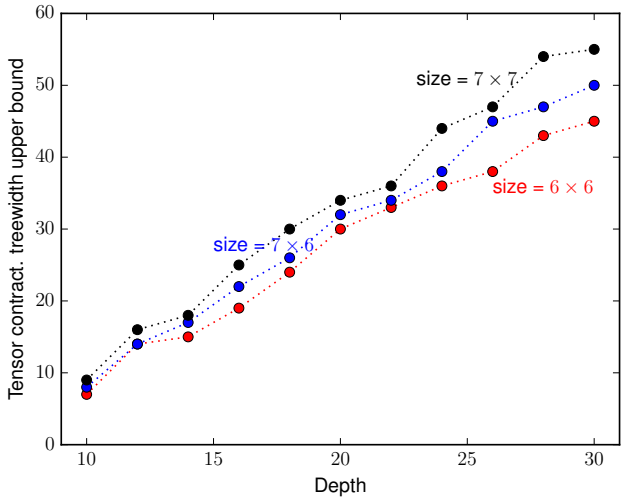

Fig. S3: Numerical upper bound for the treewidth of the line graph which gives the tensor contraction complexity [37] of circuits with 6×6, 7×6, and 7×7 qubits as a function of the circuit depth.

(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 the main text. For the cycle of two-qubit gates, we follow a similar sequence to the layouts as in the main text, but now every qubit participates in exactly one CZ gate. In addition, as mentioned above, we use periodic boundary conditions.

Figure **S2** 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. 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.

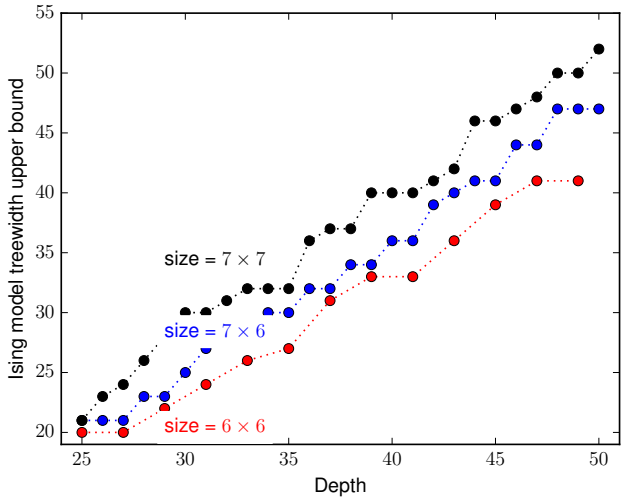

Fig. S4: Numerical upper bound for the treewidth of the interaction graph of the Ising model corresponding to circuits with 6×6, 7×6, and 7×7 qubits as a function of the circuit depth (see Sec. III B).

VII. Treewidth and simulation of low depth circuits

The output probability $p_U(x) \equiv |\langle x| U |0\rangle|^2$ for a bit-string x and quantum circuit U can be calculated using tensor contractions [37–39]. The tensor contraction complexity is exponential in the treewidth of the line graph corresponding to the circuit U [37]. Figure **S3** shows numerical upper bounds for the treewidth of the line graph as a function of depth for the circuits in the main text. The upper bounds were obtained by running the *QuickBB* algorithm [40].

As seen in Sec. III B, the amplitude for a given output bit-string can be mapped directly to the partition function of an Ising model at imaginary temperature. The partition function can be calculated exactly using the so-called variable elimination algorithm [43–45], see Ref. [41]. The cost is exponential on the treewidth of the interaction graph of the Ising model. Figure **S4** shows the treewidth calculated numerically with the *QuickBB* algorithm [40]. For a circuit in a 2D lattice of qubits with two-qubit gates restricted to nearest neighbors, the treewidth is proportional to $\min(d\sqrt{n}, n)$. Alternatively, the circuit amplitude expressed in Eq. (24) can be mapped directly to an undirected graphical model (which corresponds to the same interaction graph), and calculated by variable elimination (see Ref. [41] for details). Figure **S5** shows some indicative times per output probability amplitude on a single

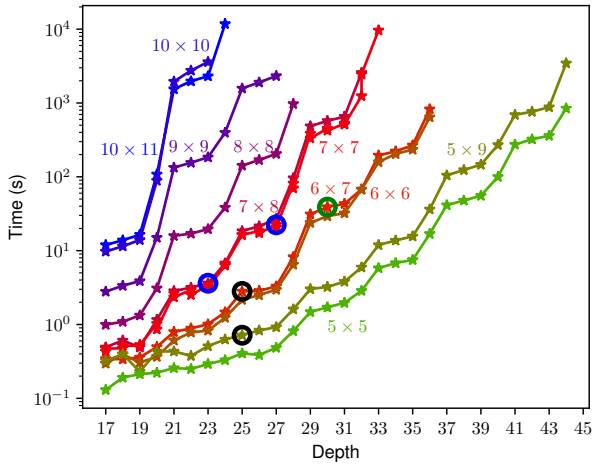

Fig. S5: Time per output probability for a typical instance as a function of depth on a single workstation, using TensorFlow as the engine for the computation, see Ref. [41]. Different colors corresponds to different circuit sizes, from 5×5 (and 5×9) qubits, to 10×10 (10×11). The green circle marker corresponds to the largest circuits simulated in this paper, black circles to Ref. [42], and blue circles to Ref. [39].

workstation, see Ref. [41]. As explained above, the cost is exponential in the depth and the smaller lateral dimension, $\sqrt{n}$ for a square lattice. This algorithm is clearly more practical for simulating sampling or cross entropy benchmarking (see Sec. II), and a supercomputer is not required. Note that circuits with 7×7 qubits and depth 40 remain out of reach. Nevertheless, this algorithm is not useful when estimating observables with exponential precision, as in Figs. **S1**, **S2** and 2c. We also note that Refs. [39, 42] use optimizations which are not useful when calculating observables as a function of depth, as done in this paper to study the convergence to Porter-Thomas.

We focus on the treewidth because it's directly related to the cost of simulating a quantum circuit [37, 39, 41], and can be estimated numerically [40]. Using a different approach, it is possible to map the partition function of an Ising model to the overlap of a product state and a so-called graph state [46]. There is some overhead in this mapping, because the graph of the graph state is extended with respect to the interaction graph of the Ising model. There is an algorithm that can calculate a partition function in this way with cost exponential in the rank width [47] of the graph state [48]. In this respect we would like to note that the quantum circuits studied here strictly contain a 2D lattice, and that the rank width of a 2D lattice grows with the size. Furthermore, the rank width of the random graphs obtained on sparse IQP circuits [32] is linear in the number of qubits [49], and in Sec. IX we argue that these circuits are in practice easier to simulate than the universal random circuits studied here.

VIII. 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. III 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,

$$\Psi = \langle x|\psi_d\rangle = \frac{1}{\sqrt{L}} \sum_{k=0}^{K-1} M_k e^{i\frac{2\pi}{K}k} \tag{52}$$

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. (26).

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 $\Pr(\Psi|\{Q_k\})$. We will see that the result is equivalent to a cross entropy difference $\sim Q^2/(NL)$. For instance, even if we sample $Q = 10^{18}$ spin configurations this will give a cross entropy difference 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

$$\begin{aligned} \Pr(\Psi) &\propto \exp(-N\Psi\Psi^*) \\ &= \exp\left(-NL \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} p_{k_1} p_{k_2} \cos\frac{2\pi}{K}(k_1 - k_2)\right) . \end{aligned} \tag{53}$$

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

$$\sum_{j=0}^{K-1} \cos\left(\frac{2\pi}{K}(m-j)\right) v_j^\alpha = \lambda_\alpha v_m^\alpha \tag{54}$$

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

of the kernel are

$$v_j^0 = \frac{1}{\sqrt{K}} \tag{55}$$

$$v_j^\alpha = \sqrt{\frac{2}{K}} \cos\left(\frac{2\pi}{K}\alpha j\right), \quad \alpha \in [1\,..\,K/2-1] \tag{56}$$

$$v_j^{K/2} = \frac{(-1)^j}{\sqrt{K}} \tag{57}$$

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

$$\lambda_\alpha = \frac{K}{2}\left(\delta_{\alpha,1} + \delta_{\alpha,K-1}\right)\ . \tag{58}$$

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

$$\Pr(\Psi) \propto \exp(-(c_1^2 + c_{K-1}^2)) \tag{59}$$

and

$$p_j = \frac{2}{K}\sqrt{\frac{1}{LN}}\left(c_1 \cos\left(\frac{2\pi}{K}j\right) + c_{K-1}\sin\left(\frac{2\pi}{K}j\right)\right) + \frac{1}{K} + \sum_{\alpha=2}^{K-2} c_\alpha v_j^\alpha\ . \tag{60}$$

We define

$$\rho_k \equiv \sum_{\alpha=2}^{K-2} c_\alpha v_k^\alpha\ . \tag{61}$$

With this definition, the numbers ρ_k obey the following constraints

$$0 = \sum_{k=0}^{K-1} \rho_k \cos\left(\frac{2\pi}{K}k\right) = \sum_{k=0}^{K-1} \rho_k \sin\left(\frac{2\pi}{K}k\right) = \sum_{k=0}^{K-1} \rho_k\ , \tag{62}$$

which will be used later to simplify the posterior probability.

The posterior probability for Ψ is

$$\Pr(\Psi|\{Q_k\}) \propto Q! \prod_{k=0}^{K-1} \frac{p_k^{Q_k}}{Q_k!} \Pr(\Psi) \tag{63}$$

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

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

$$\log \Pr\left(c_1, c_{K-1}, \{\rho_j\}|\{Q_k\}\right) \propto \sum_{j=0}^{K-1} Q_j \log\left(p_j\right) - \left(c_1^2 + c_{K-1}^2\right)\ . \tag{64}$$

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. (59). Therefore

$$\Pr(p, \{\rho_j\}|\{Q\}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Pr\left(c_1, c_{K-1}, \rho|Q\right) \delta\left(\frac{c_1^2 + c_{K-1}^2}{N} - p\right) dc_1 dc_2\ . \tag{65}$$

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

$$\Pr(p, \{\rho_j\}|\{Q_j\}) = \frac{N}{2}\int_0^{2\pi} \Pr\left(\sqrt{Np}\cos\phi, \sqrt{Np}\sin\phi, \rho|\{Q\}\right) d\phi\ . \tag{66}$$

Using Eq. (60) in Eq. (64) we write the Taylor series for the log posterior for c_1, c_{K-1} as

$$\log \Pr\left(\sqrt{Np}\cos\phi, \sqrt{Np}\sin\phi, \{\rho_j\}|\{Q_k\}\right) \propto \sum_{q=1}^{\infty} \frac{(-1)^{q+1}}{qL^{q/2}} p^{q/2} \sum_{j=0}^{K-1} Q_j \left(\frac{2\cos\left(\frac{2\pi}{K}j - \phi\right)}{K\rho_j + 1}\right)^q + \sum_{j=0}^{K-1} Q_j \log\left(\rho_j + \frac{1}{K}\right) - Np\ . \tag{67}$$

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

$$\log \Pr\left(\sqrt{Np}\cos\phi, \sqrt{Np}\sin\phi, \{\rho_j\}|\{Q_k\}\right) \propto \sqrt{\frac{p}{L}} \sum_{j=0}^{K-1} Q_j \frac{2\cos\left(\frac{2\pi}{K}j - \phi\right)}{K\rho_j + 1} + \sum_{j=0}^{K-1} Q_j \log\left(\rho_j + \frac{1}{K}\right) - Np\ . \tag{68}$$

Exponentiating we get the posterior distribution

$$\Pr\left(\sqrt{Np}\cos\phi, \sqrt{Np}\sin\phi, \{\rho_j\}|\{Q_k\}\right) \propto e^{-Np}\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\rho_j + \frac{1}{K}\right)\right) \left(1 + \sqrt{\frac{p}{L}} \sum_{j=0}^{K-1} Q_j \frac{2\cos\left(\frac{2\pi}{K}j - \phi\right)}{K\rho_j + 1} + \frac{2p}{L}\left(\sum_{j=0}^{K-1} Q_j \frac{\cos\left(\frac{2\pi}{K}j - \phi\right)}{K\rho_j + 1}\right)^2\right)\ . \tag{69}$$

Note that we keep the second term when exponentiating, which is order Q^2/L, but we drop the second term in Eq. (67), which is of order $Q/L^{3/2}$.

We can carry out a further simplification by noticing that ρ_j, which is defined in Eq. (61) from the vector of probabilities p_j, obeys $\rho_j \ll Q$. Therefore, from the form of Eq. (69), 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. (64) subject to the constraints given in Eq. (62).

We now insert Eq. (69) into Eq. (66) and carry out the integration to obtain

$$\Pr(p|\{Q_k\}) = Ce^{-Np}\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\bar{\rho}_j + \frac{1}{K}\right)\right)$$
$$\left(1 + \frac{p}{L}\sum_{j_1,j_2=0}^{K-1} Q_{j_1}Q_{j_2}\frac{\cos\left(\frac{2\pi(j_1-j_2)}{K}\right)}{(K\bar{\rho}_{j_1}+1)\,(K\bar{\rho}_{j_2}+1)}\right). \quad (70)$$

Equation (70) 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 cross entropy difference (fidelity) α of a quantum circuit U can be modeled with

$$\rho_{\mathcal{K}} = \alpha\,|\psi_d\rangle\,\langle\psi_d| + (1-\alpha)\frac{\mathbb{1}}{N}\,. \quad (71)$$

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

$$\Pr_\alpha(p_U(x)) = N^2 e^{-Np}\left(\alpha p + \frac{1-\alpha}{N}\right)\,. \quad (72)$$

We can compare the posterior distribution, given by Eq. (70), with Eq. (72) to obtain an equivalent cross entropy difference 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

$$C\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\bar{\rho}_j + \frac{1}{K}\right)\right) = (1-\alpha)N\,. \quad (73)$$

The equation linear in p gives

$$N^2\alpha = C\exp\left(\sum_{j=0}^{K-1} Q_j \log\left(\bar{\rho}_j + \frac{1}{K}\right)\right)$$
$$\frac{1}{L}\sum_{j_1,j_2=0}^{K-1} Q_{j_1}Q_{j_2}\frac{\cos\left(\frac{2\pi(j_1-j_2)}{K}\right)}{(K\bar{\rho}_{j_1}+1)\,(K\bar{\rho}_{j_2}+1)}\,. \quad (74)$$

Solving for α we obtain

$$\alpha = \frac{1}{NL}\sum_{j_1,j_2=0}^{K-1} Q_{j_1}Q_{j_2}\frac{\cos\left(\frac{2\pi(j_1-j_2)}{K}\right)}{(K\bar{\rho}_{j_1}+1)\,(K\bar{\rho}_{j_2}+1)}\,. \quad (75)$$

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

A different approximation algorithm was introduced recently in Ref. [50]. The first step is to map an universal random circuits to an IQP circuit using techniques related to measurement-based quantum computation. The second step approximates the output distribution using the Fourier analysis algorithm for IQP circuits introduced in Ref. [32]. This algorithm requires the estimation of the Fourier components of the output distribution. Unfortunately, for a Porter-Thomas distribution, the Fourier components are exponentially close to the Fourier components of the uniform distribution over bit-strings [51]. It follows that improving over the trivial sampling approximation of guessing random bit-strings with this technique is as costly as simulating the quantum circuit exactly.

IX. IQP sampling estimates

A different approach to low-depth 2D quantum circuits designed to achieve quantum supremacy, the so-called sparse IQP circuits, was introduced in Ref. [32]. These circuits are a sparse version of the circuits in Ref. [16].

An IQP circuit [11, 16, 32] can be approximately sampled if we can calculate individual probabilities for the measurement outcomes in the computational basis. Each such probability corresponds to calculating the partition function of a complex Ising model. The number of terms in the Ising function is exponential in the number of qubits [16] (while for universal random circuits the number of terms is exponential in the number of gates). The partition function can be calculated using a Gray code so when we sum the different spin configurations we only change one spin at a time. Then we only need to calculate the energy difference corresponding to flipping a spin for each term in the partition function. We can estimate that this elementary operation takes around 1 ns (close to a clock cycle in a classical computer). For this it also helps that the Ising model is sparse. It is possible to parallelize the sum of all the terms in a partition function using a large supercomputer, which can have around 1 million cores. With these estimates, we can calculate a probability for an sparse IQP circuit with 60 qubits in around $2^{60}/10^6$ ns, which is 19.22 minutes. Through numerical simulations we have seen that to achieve a Porter-Thomas distribution with an sparse IQP circuit [32] and 60 qubits we need an average degree of around 29 (number of CZs per qubit in a sparse graph). The required depth to implement this IQP sparse circuit in a 2D lattice using shorting networks is then around $29 \times 3 \times 8 = 696$ cycles. In reality, a swap is not implemented in one cycle with current superconducting qubits. Even then, this is more than an order magnitude more than the depth required for a random circuit.

Therefore it requires an order magnitude improvement in the error rates of the experimental quantum gates, which unfortunately makes it currently unfeasible.

X. 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 [52, 53]. 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, multithreading, 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 1.72 seconds. These are the largest quantum circuits simulated to-date for a computational task that approaches quantum supremacy.

A. Background

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

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

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

$$\begin{aligned} \alpha'_{*\ldots*0_k*\ldots*} &= u_{11} \cdot \alpha_{*\ldots*0_k*\ldots*} + u_{12} \cdot \alpha_{*\ldots*1_k*\ldots*} \\ \alpha'_{*\ldots*1_k*\ldots*} &= u_{21} \cdot \alpha_{*\ldots*0_k*\ldots*} + u_{22} \cdot \alpha_{*\ldots*1_k*\ldots*} \end{aligned} \tag{76}$$

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.

B. Implementation and Optimization

The implementation of single- and two-qubit controlled gates follows directly Eq. 76. For example, to apply a single-qubit gate to qubit k, we iterate over consecutive groups of amplitudes of length 2^{k+1}, applying $U_{\rm sq}$ 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 [54]. 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.[61]

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 [55]. 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 [52].

Multi-node Implementation: Single node quantum simulation is limited by the size of the physical memory of the compute node.[62] 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 2^{n-p} amplitudes. Given single- or controlled two-qubit gate operations on the target qubit k, if $k < n - p$, the operation is fully contained within a node; otherwise it requires inter-node communication. Our communication scheme follows [56], 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 [56] 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 [53, 57]. To further reduce the runtime 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 recog-

nizing 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.

C. Performance

We performed quantum simulations on Edison supercomputer [58]. 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 [55] 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 [59]. Thus the expected slowdown of gates that require communication, compared to gates that do not, is $\sim 8\times$.

Figure **S6** 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 $\sim$0.82 seconds per gate. This corresponds to 42 GB/s memory bandwidth (2 [accesses (read/write)]$\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 three-fold 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. [59]. 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.

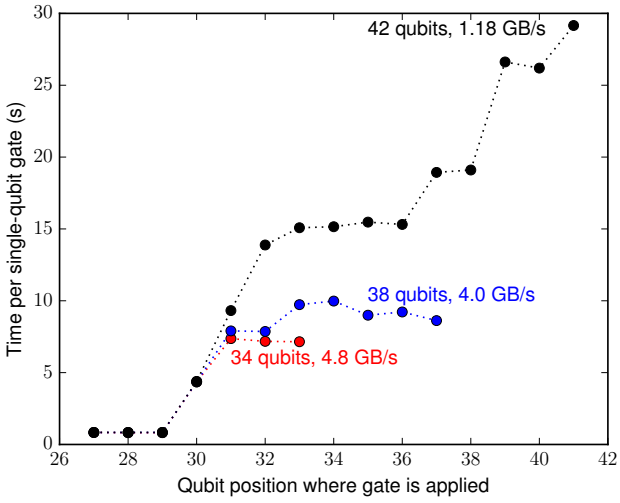

Fig. S6: Gate benchmarking results on multiple nodes (sockets) for the single-qubit Hadamard gate. The x-axis is the position of the qubit where the gate is applied. Operations on qubits in position 30 and above require network communication. The magnitude of the jump in the time per gate after position 30 is commensurate with the ratio between network and memory bandwidth. Numbers in the labels show achieved bandwidth for the higher ordered qubits.

Table **S1** 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 X B, 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

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 S1: 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. X B). The last two columns show average time per gate and time per circuit with depth 25, respectively.

also reduces compute requirements by as much as $4\times$: 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 $2.1\times$ and $1.6\times$, 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 **S6** 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: $1.4\times$ 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 $1.4\times$ 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 $2\times$.

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 $2\times$, while the time per gate has also increased, as shown in Figure **S6**. As a result, the average time per gate on a 42-qubit simulation is 1.72 seconds; a $2.3\times$ 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 $\times$ 23.0 gates per level $\times$ 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.

An improved implementation of a quantum circuit simulator was recently reported in Ref. [42] after this paper appeared in the arXiv. Ref. [42] obtains an order of magnitude speedup against the benchmarking reported here for circuits with 42 qubits, and reports simulations of circuits with 45 qubits. Nev-

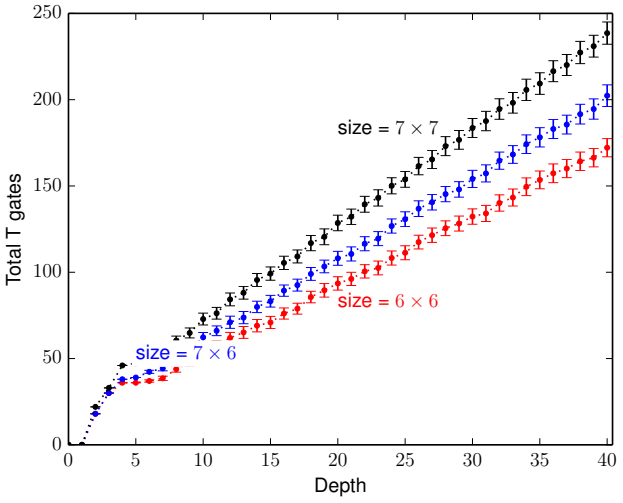

Fig. S7: Number of non-Clifford T gates as a function of depth for circuits with 6×6, 7×6, and 7×7 qubits. Error bars are the standard deviations among random circuit instances.

ertheless, as done in this paper, if we want to obtain statistics of the final state at each cycle of the quantum circuit for scientific purposes, the relative speedup will be substantially diminished.

XI. Non-Clifford gates

Clifford circuits (circuits which only contain Clifford gates) can be simulated efficiently [26]. Furthermore, this method can be extended to simulate circuits which are dominated by Clifford gates [60]. The only non-Clifford gate employed on the circuits we have used, as defined in the main text, is the T gate. Figure **S7** 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 40. 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.

[1] Haake, F. Signatures of quantum chaos (1991).

[2] Porter, C. & Thomas, R. Fluctuations of nuclear reaction widths. *Phys. Rev.* **104**, 483 (1956).

[3] Brody, T. A. *et al.* Random-matrix physics: spectrum and strength fluctuations. *Rev. Mod. Phys.* **53**, 385 (1981).

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

[5] Ullah, N. Invariance hypothesis and higher correlations of hamiltonian matrix elements. *Nuclear Physics* **58**, 65–71 (1964).

[6] Petz, D. & Rffy, J. On asymptotics of large Haar distributed unitary matrices. *Period. Math. Hungar.* **49**, 103–117 (2004).

[7] Neill, C. *et al.* A blueprint for demonstrating quantum supremacy with superconducting qubits. *arXiv:1709.06678* (2017).

[8] Aaronson, S. Quantum lower bound for recursive fourier sampling. *QIC* **3**, 165–174 (2003).

[9] Terhal, B. M. & DiVincenzo, D. P. Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games. *QIC* **4**, 134–145 (2004).

[10] Aaronson, S. Quantum computing, postselection, and probabilistic polynomial-time. In *Proc. Roy. Soc. London Ser. A*, vol. 461, 3473–3482 (2005).

[11] Bremner, M. J., Jozsa, R. & Shepherd, D. J. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. In *Proc. Roy. Soc. London Ser. A*, vol. 467, 459–472 (2011).

[12] Aaronson, S. & Arkhipov, A. The computational complexity of linear optics. In *STOC*, 333–342 (ACM, 2011).

[13] Aaronson, S. The equivalence of sampling and searching. *TOCS* **55**, 281–298 (2014).

[14] Fujii, K. *et al.* Impossibility of classically simulating one-clean-qubit computation. *arXiv:1409.6777* (2014).

[15] Jozsa, R. & Van Den Nest, M. Classical simulation complexity of extended Clifford circuits. *QIC* **14**, 633–648 (2014).

[16] Bremner, M. J., Montanaro, A. & Shepherd, D. J. Average-case complexity versus approximate simulation of commuting quantum computations. *Phys. Rev. Lett.* **117**, 080501 (2016).

[17] Farhi, E. & Harrow, A. W. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. *arXiv:1602.07674* (2016).

[18] Cook, S. A. The complexity of theorem-proving procedures. In *STOC*, 151–158 (ACM, 1971).

[19] Stockmeyer, L. The complexity of approximate counting. In *STOC*, 118–126 (ACM, 1983).

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

[21] Fujii, K. & Morimae, T. Commuting quantum circuits and complexity of Ising partition functions. *New J. Phys.* **19**, 033003 (2017).

[22] Goldberg, L. A. & Guo, H. The complexity of approximating complex-valued Ising and Tutte partition functions. *arXiv:1409.5627* (2014).

[23] Lidar, D. A. On the quantum computational complexity of the Ising spin glass partition function and of knot invariants. *New J. Phys* **6**, 167 (2004).

[24] Geraci, J. & Lidar, D. A. Classical Ising model test for quantum circuits. *New J. Phys* **12**, 075026 (2010).

[25] De las Cuevas, G., Van den Nest, M., Martin-Delgado, M. *et al.* Quantum algorithms for classical lattice models. *New J. Phys.* **13**, 093021 (2011).

[26] Gottesman, D. The heisenberg representation of quantum computers. *arXiv:quant-ph/9807006* (1998).

[27] Goldreich, O. Computational complexity: a conceptual perspective. *ACM SIGACT News* **39**, 35–39 (2008).

[28] Impagliazzo, R., Levin, L. A. & Luby, M. Pseudo-random generation from one-way functions. *STOC* 12–24 (1989).

[29] Brandao, F. G., Harrow, A. W. & Horodecki, M. Local random quantum circuits are approximate polynomial-designs. *Commun. Math. Phys.* **346**, 397–434 (2016).

[30] Hangleiter, D., Bermejo-Vega, J., Schwarz, M. & Eisert, J. Anti-concentration theorems for schemes showing a quantum

computational supremacy. *arXiv:1706.03786* (2017).

[31] Nakata, Y., Hirche, C., Koashi, M. & Winter, A. Efficient unitary designs with nearly time-independent hamiltonian dynamics. *arXiv:1609.07021* (2016).

[32] Bremner, M. J., Montanaro, A. & Shepherd, D. J. Achieving quantum supremacy with sparse and noisy commuting quantum computations. *arXiv:1610.01808* (2016).

[33] Barends, R. *et al.* Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature* **508**, 500–503 (2014).

[34] Barends, R. *et al.* Digital quantum simulation of fermionic models with a superconducting circuit. *Nat. Comm.* **6**, 7654 (2015).

[35] Kelly, J. *et al.* State preservation by repetitive error detection in a superconducting quantum circuit. *Nature* **519**, 66–69 (2015).

[36] Barends, R. *et al.* Digitized adiabatic quantum computing with a superconducting circuit. *Nature* **534**, 222–226 (2016).

[37] Markov, I. L. & Shi, Y. Simulating quantum computation by contracting tensor networks. *SICOMP* **38**, 963–981 (2008).

[38] Aaronson, S. & Chen, L. Complexity-theoretic foundations of quantum supremacy experiments. *arXiv:1612.05903* (2016).

[39] Pednault, E. *et al.* Breaking the 49-qubit barrier in the simulation of quantum circuitsn. *arXiv:1710.05867* (2017).

[40] Gogate, V. & Dechter, R. A complete anytime algorithm for treewidth. In *Proc CUAI*, 201–208 (2004).

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

[42] Häner, T. & Steiger, D. S. 0.5 Petabyte Simulation of a 45-Qubit Quantum Circuit. *arXiv:1704.01127* (2017).

[43] Dechter, R. Bucket elimination: A unifying framework for probabilistic inference. In *Learning in graphical models*, 75–104 (Springer, 1998).

[44] Kask, K., Dechter, R., Larrosa, J. & Cozman, F. Bucket-tree elimination for automated reasoning. *Artif. Intel* **125**, 131 (2001).

[45] Murphy, K. P. *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series (MIT Press, Cambridge, MA, 2012).

[46] Van den Nest, M., Dür, W. & Briegel, H. J. Completeness of the classical 2d ising model and universal quantum computation. *Phys. Rev. Lett.* **100**, 110501 (2008).

[47] Oum, S.-i. & Seymour, P. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B* **96**, 514–528 (2006).

[48] Hiraishi, H. & Imai, H. Bdd operations for quantum graph states. In *International Conference on Reversible Computation*, 216–229 (Springer, 2014).

[49] Lee, C., Lee, J. & Oum, S.-i. Rank-width of random graphs. *Journal of Graph Theory* **70**, 339–347 (2012).

[50] Yung, M.-H. & Gao, X. Can chaotic quantum circuits maintain quantum supremacy under noise? *arXiv:1706.08913* (2017).

[51] Boixo, S., Smelyanskiy, V. N. & Neven, H. Fourier analysis of sampling from noisy chaotic quantum circuits. *arXiv:1708.01875* (2017).

[52] Smelyanskiy, M., Sawaya, N. P. D. & Aspuru-Guzik, A. qHiPSTER: The Quantum High Performance Software Testing Environment (2016). arXiv:1601.07195.

[53] Häner, T., Steiger, D. S., Smelyanskiy, M. & Troyer, M. High Performance Emulation of Quantum Circuits (2016). arxiv:1604.06460.

[54] New Instruction Descriptions Now Available. Software.intel.com. Retrieved: 2012-01-17.

[55] OpenMP Architecture Review Board. OpenMP application program interface version 3.0 (2013).

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

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

[58] Edison Cray XC30. www.nersc.gov/systems/edison-cray-xc30. Accessed: 2016-04-29.

[59] Austin, B., Cordery, M., Wasserman, H. & Wright, N. Performance measurements of the NERSC Cray Cascade system. *Cray Inc.,* (2013).

[60] Bravyi, S. & Gosset, D. Improved classical simulation of quantum circuits dominated by Clifford gates. *arXiv:1601.07601* (2016).

[61] 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.

[62] 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.