Editors' Suggestion

Solving the Sampling Problem of the Sycamore Quantum Circuits

Feng Pan,[1,2] Keyang Chen,[1,3] and Pan Zhang[1,4,5,*]

[1]CAS Key Laboratory for Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Yuanpei College, Peking University, Beijing 100871, China
[4]School of Fundamental Physics and Mathematical Sciences, Hangzhou Institute for Advanced Study, UCAS, Hangzhou 310024, China
[5]International Centre for Theoretical Physics Asia-Pacific, Beijing/Hangzhou, China

(Received 20 November 2021; accepted 20 July 2022; published 22 August 2022)

We study the problem of generating independent samples from the output distribution of Google's Sycamore quantum circuits with a target fidelity, which is believed to be beyond the reach of classical supercomputers and has been used to demonstrate quantum supremacy. We propose a method to classically solve this problem by contracting the corresponding tensor network just once, and is massively more efficient than existing methods in generating a large number of *uncorrelated* samples with a target fidelity. For the Sycamore quantum supremacy circuit with 53 qubits and 20 cycles, we have generated 1×10^6 *uncorrelated* bitstrings $\mathbf{s}$ which are sampled from a distribution $\hat{P}(\mathbf{s}) = |\hat{\psi}(\mathbf{s})|^2$, where the approximate state $\hat{\psi}$ has fidelity $F \approx 0.0037$. The whole computation has cost about 15 h on a computational cluster with 512 GPUs. The obtained 1×10^6 samples, the contraction code and contraction order are made public. If our algorithm could be implemented with high efficiency on a modern supercomputer with ExaFLOPS performance, we estimate that ideally, the simulation would cost a few dozens of seconds, which is faster than Google's quantum hardware.

DOI: 10.1103/PhysRevLett.129.090502

The sampling problem of quantum circuits has been proposed recently as a specific computational task to demonstrate whether programmable quantum devices can surpass the ability of classical computations, also known as *quantum supremacy* (or quantum advantage) [1–10]. As a milestone, in 2019, Google released the Sycamore quantum circuits to realize this approach for the first time [1]. The Sycamore quantum supremacy circuits contain 53 qubits and 20 cycles of unitary operations. Google has demonstrated that the noisy sampling task with fidelity $f \approx 0.002$ can be achieved experimentally using the quantum hardware in about 200 sec, while they estimated that it would take 10000 yr on modern supercomputers.

However, the computational time estimated by Google relies on a specific classical algorithm, the Schrödinger-Feynman algorithm [1,2,11], rather than a theoretical bound that applies to all possible algorithms. So, in principle, there could exist algorithms that perform much better than the algorithm used by Google, rejecting the quantum supremacy claim. Indeed, in this Letter, we provide such an algorithm based on the tensor network method.

There have been great efforts to develop more efficient classical simulation algorithms. IBM has estimated that the 53-qubit state vector of the Sycamore circuits can be stored and evolved if one could employ all the RAM and hard disks of the Summit supercomputer. However, it is apparently unrealistic to do such a numerical experiment. Recently, a variety of methods have been proposed for this problem based on computing a single amplitude or a batch of amplitudes [5,12–15] using tensor network contractions. In particular, [15] proposed contracting the corresponding tensor network 2000 times to obtain 2000 batches of amplitudes (each batch contains 64 correlated bitstrings), then sample 2000 perfect samples from the batches and mix them with 998000 random bitstrings to obtain samples with linear cross entropy benchmark (XEB) around 0.002. However the computational cost of such simulation is still too large, and the experiment has not been realized yet.

Another attempt to pass the XEB test on the Sycamore quantum supremacy circuits is the recently proposed *big-head* approach [16], which can obtain a large number of correlated samples. Using 60 GPUs for 5 days, the authors of [16] generated 1×10^6 correlated samples with XEB 0.739, passed the XEB test. We also noticed that very recent works [17,18] implemented this approach on a supercomputer, and heavily reduced the running time for obtaining a batch of correlated samples. However, if the target of the simulation is not only passing the XEB test but also satisfying the constraint of obtaining *uncorrelated* samples, as in the Sycamore experiments, then one needs to repeat the contraction thousands of times, making the computation cost unaffordable in practice. Moreover, a recent work [19] studied a particular method for obtaining high (average) XEB values but low fidelity, illustrating limitations of XEB as a measure for fidelity.

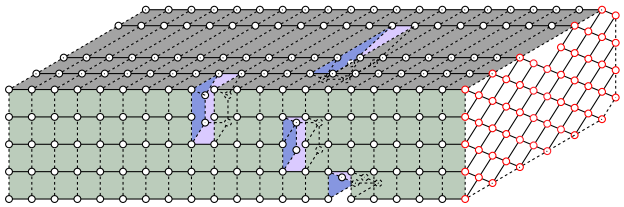

FIG. 1. Pictorial representation of the three-dimensional tensor network corresponding to the Sycamore quantum circuit with $n = 53$ qubits and $m = 20$ cycles. There are 4 holes in the tensor network designed for reducing the contraction complexity. Each hole is created by breaking two edges in a selected two-qubit gate and the companion edges, i.e., removing the entire two-qubit gate, as described in the main text. The result of contracting the three-dimensional tensor network using the sparse-state method is $L = 2^{20}$ groups of amplitudes, each group contains $l = 2^6 = 64$ correlated bitstring amplitudes. That is, we have computed approximate amplitudes for $2^{26} = 67, 108, 864$ bitstrings and finally sampled 2^{20} uncorrelated bitstrings from them.

In this Letter, we propose a tensor network approach to solve the uncorrelated sampling problem for the Sycamore quantum supremacy circuits. Our method is based on contractions of the three-dimensional tensor network $\hat{\mathcal{G}}$ (Fig. 1) converted from the quantum circuit. A single contraction of $\hat{\mathcal{G}}$ produces $\{\hat{\psi}_i^\mu\}$ with $i = 1, 2, \ldots L$ and $\mu = 1, 2, \ldots l$, representing amplitudes of L (randomly chosen) uncorrelated groups of bitstrings with each group containing l correlated bitstrings. Since $\{\hat{\psi}_i^\mu\}$ contains a small portion of entries of a approximate state $\hat{\psi}$ with fidelity F, we term it as a *sparse state*. Based on the sparse state, we do importance sampling to obtain one sample from a group, finally generating L uncorrelated samples from the approximate probability $\hat{P} = |\hat{\psi}|^2$, i.e., L approximate samples from the output distribution of the quantum circuit with fidelity F.

Our algorithm is massively more efficient than existing algorithms in generating a large number of uncorrelated samples. On the Sycamore circuits with $n = 53$ qubits and $m = 20$ cycles, we have successfully generated $L = 2^{20}$ approximate samples with fidelity $F \approx 0.0037$ in about 15 h using 512 GPUs. We remark that to the best of our knowledge this is the first time that the sampling problem of the Sycamore quantum supremacy circuits (with fidelity larger than Google's hardware samples) with $n = 53$ qubits and $m = 20$ cycles is solved in practice classically.

Method.—The quantum circuits U can be regarded as a unitary tensor network $\mathcal{G}$ with matrices (corresponding to single-qubit gates) and four-way tensors (corresponding to two-qubit gates) connecting to each other. For the Sycamore circuits where the qubits are placed on a two-dimensional layout, the corresponding $\mathcal{G}$ is a three-dimensional tensor network as illustrated in Fig. 1. The initial state (the leftmost layer) and the final state (the rightmost layer) act as two boundary conditions to $\mathcal{G}$. The initial state is always a product state so acts as a set of vectors; while the final state is represented as either a giant tensor or a set of small tensors (including vectors) depending on how many amplitudes we request in contraction of $\mathcal{G}$.

If we request all amplitudes of the final state, the final state acts as a giant tensor with size 2^n, which requires a storage space exponential to the number of qubits. If we request only one amplitude of the final state, then the boundary is a product state and acts as a set of vectors. Another case considered in the literature is the batch contraction [15,16], which requests amplitudes for l correlated bitstrings and gives a tensor with size l as the final boundary condition for $\mathcal{G}$. In this Letter our target is different: we request a large number of amplitudes for uncorrelated bitstrings, from single contraction of $\hat{\mathcal{G}}$, a slightly perturbed version of $\mathcal{G}$.

Tensor network $\hat{\mathcal{G}}$ is created by breaking (removing) K edges (connections) in $\mathcal{G}$. The edge breaking is implemented by inserting $E = \binom{1}{0} \otimes \binom{1}{0}$ in between the two tensors that the edge is connecting. In this Letter, we select K edges from input indices of $K/2$ two-qubit gates. Pictorially it represents as drilling $K/2$ holes in the three-dimensional graphical representation of $\hat{\mathcal{G}}$ as shown in Fig. 1. The position of holes are determined such that contracting $\hat{\mathcal{G}}$ is much easier than contracting $\mathcal{G}$, but with the price of decreasing the fidelity. The amount of decreased fidelity can be estimated using the expression of E as a specific Pauli error matrix $E = \frac{1}{2}I + \frac{1}{2}\sigma_z$, with $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and $\sigma_z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. The effect of the edge breaking can be understood as breaking the system into a summation of two subnetworks. The first subnetwork is a copy of the original one which preserves the information of the original final state, while the second subnetwork with the action of σ_z completely destroys the information of the original final state. Since the weight of each subnetwork is $1/2$ [11], one then estimates that each edge breaking decreases the fidelity F by a factor of $1/2$. After breaking K edges in $\mathcal{G}$, we arrive at $\hat{\mathcal{G}}$. If we contract $\hat{\mathcal{G}}$ and obtain a full amplitude state vector $\hat{\psi}$, it would be an approximation to the final state ψ of $\mathcal{G}$, with fidelity estimated as $F_K \approx 2^{-K}$.

The simulation method based on tensor network contractions can be regarded as Feynman's path-integral approach, because the tensor contractions effectively sum over an exponential number of paths which are considered to be orthogonal to each other, hence contributing equally to the obtained amplitudes. Under this viewpoint, the hole drilling in $\mathcal{G}$ can be understood as omitting some paths in the path-integral approach, summing over only a fraction of 2^{-K} paths, giving fidelity $F_K \approx 2^{-K}$.

In this Letter we only request the sparse state, the amplitudes for $L \times l$ bitstrings which are grouped into L groups with each group containing l correlated bitstring amplitudes (in the practical $L = 2^{20}$ and $l = 2^6$). They are given according to a generation process in advance and kept fixed during the contraction.

However, contracting $\hat{\mathcal{G}}$ to arrive at the $L \times l$ size sparse state is a very difficult task, and the space complexity of the contraction would be much larger than $L \times l$. To solve the problem we extend the *big-head* algorithm proposed in [16]. In the big-head algorithm, the three-dimensional tensor network is cut into two parts, $\hat{\mathcal{G}}_{\text{head}}$ whose contraction cost dominates the whole computation, and $\hat{\mathcal{G}}_{\text{tail}}$ which contains all the qubits in the final state and can completely reuse the contraction results of $\hat{\mathcal{G}}_{\text{head}}$ for computing all the requested amplitudes. In this Letter, the big-head method is extended to work with the sparse state (rather than a batch of correlated bitstrings in [16]). To this end, we need to balance the computation cost of $\hat{\mathcal{G}}_{\text{head}}$ and the cost of $\hat{\mathcal{G}}_{\text{tail}}$. The contraction results of $\hat{\mathcal{G}}_{\text{head}}$ is a vector $\mathbf{v}_{\text{head}}$, with size much larger than our storage limit, so in practice, we enumerate k entries in $\mathbf{v}_{\text{head}}$, that is, making 2^k slices of the $\mathbf{v}_{\text{head}}$, each slice has size 2^{29}. Given each slice of $\mathbf{v}_{\text{head}}$, the $\hat{\mathcal{G}}_{\text{head}}$ is contracted with a good contraction order and local dynamic slicing, similar to [16].

The boundary condition given by the sparse state is heavier to deal with than the boundary conditions of $\hat{\mathcal{G}}_{\text{head}}$. We proposed a new *zigzag* method for finding a good contraction order. The method starts at the beginning boundary of $\hat{\mathcal{G}}_{\text{tail}}$, contracting neighboring tensors in a complexity-greedy manner all the way towards the boundary of the sparse state, then turns around to contract greedily the tensors and come back to the beginning boundary. The process is repeated until all the tensors in $\hat{\mathcal{G}}_{\text{tail}}$ are contracted, and the sparse state $\{\psi_i^\mu\}$ is obtained. The spirit of the zigzag contraction order is to make use of both boundaries to reduce the space and time complexity of contraction. For more details about the head-tail splitting of the circuits, the sparse-state contraction method, and slicing technique, please refer to the Supplemental Material [20].

In the Sycamore circuits, two-qubit unitary transformations are parametrized using the fSim gates

$$\text{fSim}(\theta,\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -i\sin\theta & 0 \\ 0 & -i\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{bmatrix}. \quad (1)$$

Specifically, the parameters in Google's experiments [1] are tuned to $\theta \approx \pi/2$ in order to keep the decomposition rank equal to 4 with a near-flat spectrum, that is, the singular values of the 4×4 matrix obtained by reshaping the fSim gate are almost identical [1]. This setting significantly increases the cost of classical simulations when compared with controlled-Z gates which has decompositional rank 2, in exact simulations and in approximate simulations [21,22].

However, we observe that in our approach there are two situations that we can explore the low rank structures. (i) In the hole drilling, when the two input indices (α and β) of the fSim gate are cut, i.e., applying two Pauli errors gate as $A = (\left[\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right] \otimes \left[\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right]) \cdot \text{fSim}(\theta,\phi)$, as illustrated in Fig. 2 top. It evaluates to a rank-one matrix $B = \left[\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right]$, hence the fSim gate can be replaced by two (1,0) vectors without decreasing fidelity. (ii) In enumerating k entries of $\mathbf{v}_{\text{head}}$ as well as in the slicing process, fixing an index is regarded as breaking one input edge in the tensor diagram as illustrated in Fig. 2 bottom (e.g., the top left edge γ of tensor D is cut), giving a three-way tensor E. Although the decompositional rank of E on the bottom right index ω is 2, the corresponding squared singular values, $[\sin^2(\theta)+1, \cos^2(\theta)]$, are heavily imbalanced in the Sycamore circuits with $\theta \approx \pi/2$. In this way we can do a rank-one approximation by dropping the singular vectors corresponding to the squared singular value $\cos^2(\theta)$. This rank-one approximation decreases the fidelity approximately by a factor $[\sin^2(\theta)+1]/2$, while effectively break another edge ω, which we term as the *companion edge* in the tensor network. For total k selected slicing edges in the tensor network, we do the rank-one approximation for associated fSim gates, cutting k associated companion edges. This decreases the fidelity F by a factor $\prod_{i=1}^{k}[\sin^2(\theta_i)+1]/2$.

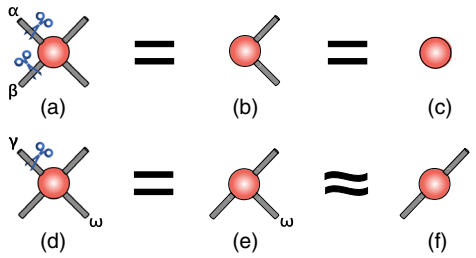

FIG. 2. Two situations where we can explore the low-rank structures. (Top) When two indices α, β are pinned to 0 for the fSim gate. The result is a rank-one matrix B effectively equals to a scalar $c = 1$ in the tensor network. (Bottom) When one index γ of an fSim gate is pinned, the resulting tensor E has decompositional rank 2 but with imbalanced singular values $[\sqrt{\sin^2(\theta)+1}, \cos(\theta)]$ with $\theta \approx \pi/2$.

After contracting $\hat{\mathcal{G}}$ using the methods we have introduced above, the sparse state $\{\hat{\psi}_i^\mu\}$ is obtained, which are selected from 2^n entries of a state $\hat{\psi}$, with fidelity to the true state estimated as $F_{\text{estimate}} \approx 2^{-K}\prod_{i=1}^{k}[\sin^2(\theta_i)+1]/2$. Since the sparse-state $\{\hat{\psi}_i^\mu\}$ is composed of L groups and each group contains l amplitudes, we use a Markov chain to sample one bitstring out of l amplitudes in each group using the Metropolis algorithm [23], producing L samples, which is considered as unbiased samples from $\hat{\psi}$. We also note that if $|\hat{\psi}|^2$ follows the Porter-Thomas distribution [8,24,25] (as we verify empirically in Fig. 3), we can use the frugal sampling [1,11], which is much faster and guaranteed to give near-perfect samples with $l = 64$. We remark that to obtain L uncorrelated samples, only groups need to be independently and randomly generated, it is not necessary to maintain uncorrelated bitstrings inside of each group [26]. The validations

of our approximate sampling method using smaller Sycamore circuits can be found in the Supplemental Material [20].

Results.—We focus on the Sycamore circuits with $n = 53$ qubits, $m = 20$ cycles, sequence *ABCDCDAB*, which have been used to demonstrate the quantum supremacy based on the estimated 10000 yr for classical simulations [1]. We first simplify the tensor network by contracting order-one and order-two tensors into their neighbors, resulting in a tensor network with $n = 455$ tensors. To arrive at $\hat{\mathcal{G}}$, we chose $K = 8$ edges to break, they are associated with 4 fSim gates. Using the low-rank structure, we completely remove the two-qubit gates by introducing proper Pauli error gates. This gives 4 holes marked in Fig. 1. This approximation decreases the fidelity by a factor 2^{-8}. Then the tensor network is divided into two parts, the head part $\hat{\mathcal{G}}_{\text{head}}$ and the tail part $\hat{\mathcal{G}}_{\text{tail}}$.

We introduce 6 slicing edges in contracting $\hat{\mathcal{G}}_{\text{head}}$. The space and time complexity are 2^{30} and 2.3816×10^{13}, respectively. Contracting the $\hat{\mathcal{G}}_{\text{head}}$ results in a tensor $\mathbf{v}_{\text{head}}$ of size 2^{45}, which we cannot store, so we enumerate 16 entries of the $\mathbf{v}_{\text{head}}$, creating 2^{16} subtasks of tensor network contraction, each of which corresponds to a configuration of 16 binary variables.

In each subtask, $\mathbf{v}_{\text{head}}$ is sliced to a tensor with size 2^{29}, which works as a boundary for $\hat{\mathcal{G}}_{\text{tail}}$. For the Sycamore circuits with $n = 53$ qubits and $m = 20$ cycles, we set $L = 2^{20}$ and $l = 2^6$, i.e., organizing the requested bitstrings to 2^{20} independent groups, each of which contains 2^6 bitstrings. It acts as another boundary of $\hat{\mathcal{G}}_{\text{tail}}$. In contracting $\hat{\mathcal{G}}_{\text{tail}}$, we introduces 7 local slicing edges, and the space and time complexity in our sparse-state contraction scheme are 2^{30} and 2.9425×10^{13}, respectively. The overall time complexity of the entire computation (for finishing 2^{16} subtasks) is 3.489×10^{18}, which is slightly lower than the previous work [16] in computing a large batch of correlated bitstring amplitudes, and [15] in computing a small batch of correlated bitstrings.

In contracting $\hat{\mathcal{G}}_{\text{tail}}$, there are 5 slicing edges associated with a companion edge. Together with the 16 companion edges in enumerating $\mathbf{v}_{\text{head}}$, there are totally $k = 21$ companion edges. We do further low-rank approximations on the $k = 21$ associated fSim gates, decreasing the fidelity by a factor $\prod_{i=1}^{21}[\sin^2(\theta_i) + 1]/2 \approx 0.9565$, where θ_i in the equation denotes the parameters of involved fSim gates. Together with the fidelity decreasing introduced in hole drilling, the final fidelity is estimated as

$$F_{\text{estimate}} = 2^{-8} \times 0.9565 \approx 0.0037. \tag{2}$$

To increase the GPU efficiency, the branch merge strategy [16,27] was adopted during the contraction. After branch merging, the GPU efficiency is 31.76% for $\hat{\mathcal{G}}_{\text{head}}$ and 14.27% for $\hat{\mathcal{G}}_{\text{tail}}$, the overall efficiency is 18.85%. We use the *Complex64* as data type in contraction. The contraction time of $\hat{\mathcal{G}}_{\text{head}}$ for one subtask is around 112 sec and that of $\hat{\mathcal{G}}_{\text{tail}}$ is around 315 sec, summing to 427 sec for completing a single subtask. The entire simulation with 2^{16} subtasks is finished in about 15 h using a computational cluster with 512 GPUs. Detailed data about the complexity, estimated fidelity, GPU efficiency are listed in the Supplemental Material [20].

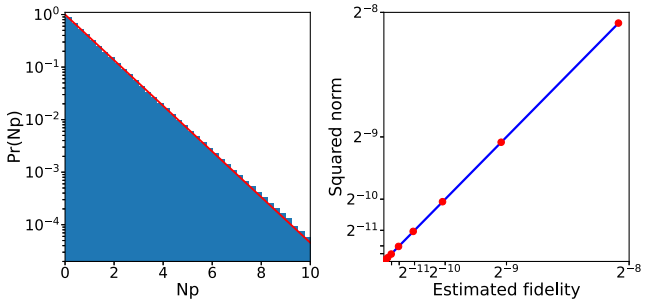

FIG. 3. Left: Histogram of approximate bitstring probabilities $p(\mathbf{s}) = |\hat{\psi}(\mathbf{s})|^2/\mathcal{N}_s$ for 2^{26} bitstrings obtained from the Sycamore circuits with $n = 53$ qubits and $m = 20$ cycles. $\mathcal{N}_s$ is the norm factor and $N = 2^n$. The estimated fidelity $\hat{\psi}(\mathbf{s})$ to the true final state $\psi(\mathbf{s})$ is $F \approx 0.0037$. The red line denotes the Porter Thomas distribution. Right: Comparison between the estimated fidelity (blue lines) and the norm factor of $\hat{\psi}_L(\mathbf{s})$ obtained by summing over a fraction of paths.

By summing over 2^{16} paths, $\hat{\mathcal{G}}$ is contracted. The results are 2^{26} bitstrings amplitudes grouped into 2^{20} uncorrelated groups corresponding to partial bitstrings $\mathbf{x} \in \{1, 0\}^{47}$ that are uniformly and randomly selected. Each group is composed of $2^6 = 64$ correlated bitstrings corresponding to 6 open qubits. As a sanity check, we compute the squared norm $\mathcal{N} = \sum_{i=1}^{2^{20}} \sum_{\mu=1}^{64} |\hat{\psi}_i^{\mu}|^2$ of the sparse-state by summing only a fraction of total paths, and compare to the expected fidelity with partial summation (i.e. the fraction of the paths). The result are shown in Fig. 3 right, where we can see that they coincide to each other. Using the norm of the sparse state we can estimate the normalization factor of the approximate distribution as $2^{27}\mathcal{N}$, and compute the approximate probability of bitstrings. The histogram of the probability is plotted in Fig. 3 left, where we can see that it fits very well to the Porter-Thomas distribution.

Finally, we generate 2^{20} uncorrelated bitstrings from the distribution of the sparse state using the MCMC importance sampling. The other method that we have tried is the frugal sampling, which is guaranteed to work well [1,11] as the distribution fits to the Porter-Thomas distribution.

Discussions.—We have presented a tensor network method for solving the approximate (uncorrelated) sampling problem of the Sycamore quantum circuits which was thought to be impossible for classical computations. Using our algorithm the simulation for the Sycamore circuits with $n = 53$ qubits and $m = 20$ cycles is completed in about 15 h using 512 V100 GPUs. There are several places that the proposed algorithms can be further speed up. First,

our contraction algorithm is straightforwardly implemented using PYTORCH. We expect that using a library that is more suitable for tensor contractions, such as the cuQuantum [28], the computational efficiency can be greatly increased. Second, in recent days a modern supercomputer could achieve a performance of ExaFLOPS (10^{18} floating-point operations per second). If our simulation of the quantum supremacy circuits (with about 2.79×10^{19} floating-point operations without branch merging) can be implemented in a modern supercomputer with high efficiency, in principle, the overall simulation time can be reduced to a few dozens of seconds, which is faster than Google's hardware experiments.

The Sycamore circuit files are retrieved from [29], and the circuits are loaded with Cirq [30] script contained in the data repository and converted to the tensor network $\mathcal{G}$. Our contraction code is implemented using *Pytorch* (version 1.7.2) with *cudatoolkit* (version 10.1). The samples and the contraction code together with the contraction orders and slicing indices for reproducing our results are available at [31]. The computation was carried out at the Cloud Brain I Computing Facility at the Peng Cheng Laboratory and HPC cluster of ITP, CAS.

We acknowledge Pengxiang Xu for help in performing computations on the Cloud Brain computers. We also thank Xun Gao, Ying Li, Lei Wang, Song Cheng, and Xiao Yuan for helpful discussions. P. Z. was supported by Chinese Academy of Sciences Grant No. QYZDB-SSW-SYS032, and Projects No. 11747601 and No. 11975294 of National Natural Science Foundation of China.

*panzhang@itp.ac.cn

[1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[2] Scott Aaronson and Lijie Chen, Complexity-theoretic foundations of quantum supremacy experiments, arXiv:1612.05903.

[3] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani, On the complexity and verification of quantum random circuit sampling, Nat. Phys. **15**, 159 (2019).

[4] Ramis Movassagh, Quantum supremacy and random circuits, arXiv:1909.06210.

[5] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, and Hartmut Neven, Simulation of low-depth quantum circuits as complex undirected graphical models, arXiv:1712.05384.

[6] Scott Aaronson and Sam Gunn, On the classical hardness of spoofing linear cross-entropy benchmarking, arXiv:1910.12085.

[7] Alexander Zlokapa, Sergio Boixo, and Daniel Lidar, Boundaries of quantum supremacy via random circuit sampling, arXiv:2005.02464.

[8] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven, Characterizing quantum supremacy in near-term devices, Nat. Phys. **14**, 595 (2018).

[9] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan *et al.*, Strong Quantum Computational Advantage Using a Superconducting Quantum Processor, Phys. Rev. Lett. **127**, 180501 (2021).

[10] Qingling Zhu, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, Ming Gong *et al.*, Quantum computational advantage via 60-qubit 24-cycle random circuit sampling, Sci. Bull. **67**, 240 (2022).

[11] Igor L. Markov, Aneeqa Fatima, Sergei V. Isakov, and Sergio Boixo, Quantum supremacy is both closer and farther than it appears, arXiv:1807.10749.

[12] Jianxin Chen, Fang Zhang, Mingcheng Chen, Cupjin Huang, Michael Newman, and Yaoyun Shi, Classical simulation of intermediate-size quantum circuits, arXiv:1805.01450.

[13] Chu Guo, Yong Liu, Min Xiong, Shichuan Xue, Xiang Fu, Anqi Huang, Xiaogang Qiang, Ping Xu, Junhua Liu, Shenggen Zheng, He-Liang Huang, Mingtang Deng, Dario Poletti, Wan-Su Bao, and Junjie Wu, General-Purpose Quantum Circuit Simulator with Projected Entangled-Pair States and the Quantum Supremacy Frontier, Phys. Rev. Lett. **123**, 190501 (2019).

[14] Johnnie Gray and Stefanos Kourtis, Hyper-optimized tensor network contraction, Quantum **5**, 410 (2021).

[15] Cupjin Huang, Fang Zhang, Michael Newman, Junjie Cai, Xun Gao, Zhengxiong Tian, Junyin Wu, Haihong Xu, Huanjun Yu, Bo Yuan *et al.*, Classical simulation of quantum supremacy circuits, arXiv:2005.06787.

[16] Feng Pan and Pan Zhang, Simulation of Quantum Circuits Using the Big-Batch Tensor Network Method, Phys. Rev. Lett. **128**, 030501 (2022).

[17] Haohuan Fu, Yuling Yang, Jiawei Song, Pengpeng Zhao, Zhen Wang, Dajia Peng, Huarong Chen, Chu Guo, Heliang Huang, Wenzhao Wu *et al.*, Closing the" quantum supremacy" gap: Achieving real-time simulation of a random quantum circuit using a new sunway supercomputer, arXiv:2110.14502.

[18] Xin Liu, Chu Guo, Yong Liu, Yuling Yang, Jiawei Song, Jie Gao, Zhen Wang, Wenzhao Wu, Dajia Peng, Pengpeng Zhao, Fang Li, He-Liang Huang, Haohuan Fu, and Dexun Chen, Redefining the quantum supremacy baseline with a new generation sunway supercomputer, arXiv:2111.01066.

[19] Xun Gao, Marcin Kalinowski, Chi-Ning Chou, Mikhail D. Lukin, Boaz Barak, and Soonwon Choi, Limitations of linear cross-entropy as a measure for quantum advantage, arXiv:2112.01657.

[20] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.129.090502 for details about the explicit head-tail splitting of the circuits, the sparse-state contraction method, slicing technique, validations of the approximate sampling method and detailed data about the complexity, estimated fidelity, and GPU efficiency.

[21] Yiqing Zhou, E. M. Stoudenmire, and Xavier Waintal, What Limits the Simulation of Quantum Computers?, Phys. Rev. X **10**, 041038 (2020).

[22] Feng Pan, Pengfei Zhou, Sujie Li, and Pan Zhang, Contracting Arbitrary Tensor Networks: General Approximate Algorithm and Applications in Graphical Models and Quantum Circuit Simulations, Phys. Rev. Lett. **125**, 060503 (2020).

[23] M. Newman and G. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, New York, USA (1999), Chap. 1-4.

[24] Charles E. Porter and Robert G. Thomas, Fluctuations of nuclear reaction widths, Phys. Rev. **104**, 483 (1956).

[25] Tómas A. Brody, Jorge Flores, J. Bruce French, P. A. Mello, A. Pandey, and Samuel S. M. Wong, Random-matrix physics: Spectrum and strength fluctuations, Rev. Mod. Phys. **53**, 385 (1981).

[26] Benjamin Villalonga, Sergio Boixo, Bron Nelson, Christopher Henze, Eleanor Rieffel, Rupak Biswas, and Salvatore Mandrà, A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware, npj Quantum Inf. **5**, 1 (2019).

[27] Cupjin Huang, Fang Zhang, Michael Newman, Xiaotong Ni, Dawei Ding, Junjie Cai, Xun Gao, Tenghui Wang, Feng Wu, Gengyan Zhang, Hsiang-Sheng Ku, Zhengxiong Tian, Junyin Wu, Haihong Xu, Huanjun Yu, Bo Yuan, Mario Szegedy, Yaoyun Shi, Hui-Hai Zhao, Chunqing Deng, and Jianxin Chen, Efficient parallelization of tensor network contraction for simulating quantum computation, Nat. Comput. Sci. **1**, 578 (2021).

[28] https://developer.nvidia.com/cuquantum-sdk.

[29] John M. Martinis *et al.*, Quantum supremacy using a programmable superconducting processor, Dryad, Dataset, 10.5061/dryad.k6t1rj8 (2021).

[30] Cirq Developers, Cirq, See full list of authors on Github: https://github.com/quantumlib/Cirq/graphs/contributors (2021).

[31] https://github.com/Fanerst/solve_sycamore.