Comparison of quantum advantage experiments using random circuit sampling

Sangchul Oh and Sabre Kais*

Department of Chemistry, Department of Physics and Astronomy, and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, Indiana, USA

(Received 23 November 2022; accepted 31 January 2023; published 13 February 2023)

Random circuit sampling, the task of sampling bit strings from a random unitary operator, has been implemented to demonstrate quantum advantage on the Sycamore quantum processor with 53 qubits and on the Zuchongzhi quantum processor with 56 and 61 qubits. Recently, it was claimed that classical computers using tensor network simulation could catch on to current noisy quantum processors for random circuit sampling. While the linear cross-entropy benchmark fidelity was used to certify all these claims, it may not capture statistical properties of outputs in detail. Here, we compare the bit strings sampled from classical computers using tensor network simulation by Pan *et al.* [F. Pan, K. Chen, and P. Zhang, Phys. Rev. Lett. **129**, 090502 (2022)] and by Kalachev *et al.* [G. Kalachev, P. Panteleev, P. Zhou, and M.-H. Yung, arXiv:2112.15083] with the bit strings from the Sycamore quantum processor. It is shown that all of Kalachev *et al.*'s samples passed the NIST random number tests. The heat maps of bit strings show that Pan *et al.*'s and Kalachev *et al.*'s samples are quite different from the Sycamore or Zuchongzhi samples. The analysis with the Marchenko-Pastur distribution and the Wasssertein distances demonstrates that Kalachev *et al.*'s samples are statistically closer to the Sycamore samples than Pan *et al.*'s while the three datasets have similar values for the linear cross-entropy fidelity. Our finding implies that further study is needed to certify or beat the claims of quantum advantage using random circuit sampling.

DOI: 10.1103/PhysRevA.107.022610

I. INTRODUCTION

Quantum advantage or quantum supremacy, the outperformance of quantum computers over classical computers for some computational tasks, is one of the main subjects of quantum computing research [1–3]. Random quantum circuit sampling [4–6] and Boson sampling [7,8] are considered good computational tasks for demonstrating quantum advantage with current noisy quantum computers. In 2019, the Google team claimed to have achieved quantum advantage for random circuit sampling [9]. The Google Sycamore quantum processor with 53 qubits took 200 seconds to sample ten millions of bit-strings from a random quantum circuit while a supercomputer at that time was estimated to take 10,000 years using the Schrödinger-Feynman algorithm to simulate the random quantum circuits [9]. Subsequently, Wu *et al.* [10] and Zhu *et al.* [11] implemented random circuit sampling on the Zuchongzhi quantum processor with 56 and 61 qubits, respectively. Also, quantum advantage for Boson sampling with optical qubits was reported [12,13].

On the opposite side, to close the quantum advantage gap, the tensor network simulation of random quantum circuits was performed on classical computers [14–20]. For example, Kalachev *et al.* [17] reported that a tensor-contraction algorithm on a graphic processing unit took 14 days to sample millions of bit strings from the same random circuit that was implemented on the Sycamore processor with 53 qubits and 20 cycles. Similarly, Pan *et al.* [19,20] showed that graphic processing units using a tensor network took 1 day to sample millions of bit strings from the same random quantum circuit performed on the Sycamore processor with 53 qubits and 20 cycles. Thus, it was claimed that classical computers could catch up to the current noisy quantum processors [21].

All of above arguments on quantum advantage for random circuit sampling on quantum or classical processors are based on two metrics: computational time and the linear XEB (cross-entropy benchmark) fidelity. The first metric measures the time it takes to accomplish a computational task and may be a primary measure to compare the performance of computers. The second metric is employed to ensure that the output of noisy quantum processors is close to an ideal result. The linear XEB is zero if the bit strings are sampled from a classical uniform probability distribution. While a nonzero value of the linear XEB was used to support the claims of quantum advantage for random circuit sampling, the linear XEB has some limitations in addition to the intrinsic scalability issue [6]. The linear XEB could be spoofed without fully simulating a random quantum circuit [22,23]. Our previous works [24,25] demonstrated that the linear XEB cannot capture the statistical properties of samples in the sense that the Sycamore bit strings are statistically different from the Zuchongzhi bit strings while both processors show similar linear XEB values as a function of the number of qubits or the number of cycles.

While the benchmark and certification of quantum processors are important to ensure the faithful calculation with a quantum processor, they are not fully explored [26]. To validate the claim of quantum advantage and to verify quantum circuits, it would be better to directly compare the statistics of outcomes generated by quantum or classical processors. The

*kais@purdue.edu

quantum advantage regime means the verification with classical processors is very inefficient. In this paper, we compare bit strings sampled from the random circuits on the Sycamore processor [27] and those calculated from tensor network simulation of the Sycamore random circuit on classical computers by Kalachev *et al.* [28] and by Pan *et al.* [29]. The paper is organized as follows. In Sec. II, we perform the NIST random number tests on bit strings and plot the heat maps of bit strings. In addition, we calculate the Marchenko-Pastur distances and the Wasserstein distances between bit strings. These statistical analyses will show that the samples from classical simulations are quite different from those from the Sycamore quantum processor. In Sec. III, we summarize our results and discuss the recent claims of quantum advantage for random circuit sampling on a quantum processor or on a classical computer.

II. SAMPLING FROM RANDOM CIRCUITS

A. Random quantum circuit sampling

Let us start with a brief summary of random quantum circuit sampling. The purpose of random quantum circuit sampling is to sample n-bit strings, $x \equiv a_1 a_2 \cdots a_n$, with $a_i = 0, 1$ and $i = 1, \ldots, n$ from a probability $p(x) = |\langle x|U|0^{\otimes n}\rangle|^2$ defined by a random unitary operator U. Here, $|x\rangle = |a_1 a_2 \cdots a_n\rangle$ stands for a computational basis of n qubits and $|0^{\otimes n}\rangle = |00\cdots 0\rangle$ for an initial state. In an ideal situation, a random unitary operator U is sampled uniformly and randomly from the unitary group $\mathbb{U}(N)$ with $N = 2^n$. A unique probability measure that is invariant under group multiplication is known as the Haar measure. A set of random unitary matrices drawn from $\mathbb{U}(N)$ according to the Haar measure is called the circular unitary ensemble, which was introduced by Dyson [30].

A random unitary matrix with respect to the Haar measure can be drawn using the Euler angle method [31–33] or the QR decomposition algorithm [34]. Hurwitz [31] showed that a unitary matrix can be constructed through the parametrization of N^2 elements with Euler angles. The QR decomposition algorithm to generate a random unitary matrix U can be done through the QR decomposition of a matrix Z with complex normal random entries $Z = QR$ where Q is a unitary matrix and R is an upper-triangle matrix. Then $U = Q\Lambda$ is a Haar-measure random unitary where Λ is the diagonal matrix with diagonal entries $\Lambda_{ii} = R_{ii}/|R_{ii}|$ and R_{ii} are the diagonal entries of the upper-triangle matrix R [34].

Emerson *et al.* [35] proposed the implementation of pseudorandom unitary operators on a quantum computer by applying single-qubit gates on each qubit and simultaneous Ising interaction on all qubits. The Google and USTC experiments implemented a pseudorandom unitary operator that is composed of m cycles of the sequence of n single-qubit gates selected randomly from $\{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$ and fixed two-qubit gates for according to tiling patterns.

A random unitary operator U transforms the initial state $|0^{\otimes n}\rangle$ into a random quantum state

$$|\psi\rangle = U|0^{\otimes n}\rangle = \sum_{x=0}^{2^n-1} \sqrt{p_x}\, e^{i\theta_x}|x\rangle, \tag{1}$$

where $p_x = p(x)$. It is known that the probability distribution Pr of p_x with $x = 0, \ldots, 2^n - 1$ is given by $Pr(p) = (N-1)(1-p)^{N-2}$ with $N = 2^n$, the eigenvector distribution of the circular unitary ensemble [30]. For $N \gg 1$, it becomes $Pr(p) \approx N\exp(-Np)$ and is known as the χ^2 distribution with 2 degrees of freedom, or the exponential distribution [36–38]. The output of random circuit sampling is a set of M bit strings, $\mathcal{D} = \{x_1, x_2, \ldots, x_M\}$, generated by M measurements of $|\psi\rangle$ in computational basis. Note that a current quantum processor could generate only a bit-strings x, but not an amplitude $\sqrt{p_x}\, e^{i\theta_x}$, while a classical computer could calculate both quantities. The phases θ_x are distributed uniformly in the range $[-\pi, \pi]$.

Today quantum processors are noisy and imperfect so the output of quantum computation could be deviated from an ideal solution. It is important to verify whether quantum computation is performed faithfully [26]. Gilchrist *et al.* [39] suggested a golden standard for the distance measure between ideal and noisy quantum processors. For sampling computation where the task is to sample outcomes x from an ideal probability distribution $p(x)$, Gilchrist *et al.* proposed the Kolmogorov distance $D(p, q) \equiv \sum_x |p(x) - q(x)|/2$ and the Bhattacharya overlap $F(p, q) \equiv \sum_x \sqrt{p(x)q(x)}$ where $p(x)$ is an ideal distribution and $q(x)$ is a real probability distribution generated by a quantum processor. On the other hand, Arute *et al.* introduced the linear XEB

$$F_{\mathrm{XEB}} = \frac{2^n}{M}\sum_{i=1}^{M} p_{\mathrm{ideal}}(x_i) - 1, \tag{2}$$

where the set of bit strings, $\mathcal{D} = \{x_1, x_2, \ldots, x_M\}$, is the measurement outcomes of a noisy quantum processor and $p_{\mathrm{ideal}}(x) = |\langle x|U|0\rangle^{\otimes n}|^2$ is an ideal probability of a quantum circuit U. For a classical uniform random distribution $p_{\mathrm{ideal}} = 1/2^n$, the linear XEB becomes zero. Basically, the linear XEB is derived from the cross entropy between the ideal distribution $p_{\mathrm{ideal}}(x)$ and the real distribution $p_{\mathrm{real}}(x)$ or the Kullback-Leibler divergence. However, the Kullback-Leibler divergence is not a true metric because it is not symmetric and does not satisfy the triangular inequality.

One of the main challenges in calculating the linear XEB or the Kolmogorov distance is how to obtain the ideal probability distribution $p_{\mathrm{ideal}}(x)$ or the real probability distribution $p_{\mathrm{real}}(x)$. For sampling calculation, a quantum computer produces bit strings rather than $p(x)$. In the quantum advantage regime, it is hard or inefficient for a classical computer to calculate $p(x)$ from quantum simulation. Also it is impossible to estimate the empirical probability distribution from the measurement data of a quantum processor since the number of measurements M is much smaller than 2^n. In the Google experiment with the Sycamore quantum processor, the number of measurements M are in the order of 10^7 but the dimension of the Hilbert space of 53 qubits is $2^{53} \sim 9 \times 10^{15}$. Thus, to verify the performance of quantum computers in the quantum advantage regime, it would be better to directly compare two data sets, $\mathcal{D}_1 = \{x_1, x_2, \ldots, x_M\}$ and $\mathcal{D}_2 = \{y_1, y_2, \ldots, y_M\}$, without calculating the probability distributions $p(x)$ or $q(x)$. For example, since classical uniform random bit strings give rise to zero of the linear XEB, they could be one reference point.

TABLE I. Datasets of random circuit sampling. The datasets of Sycamore [27], Kalachev *et al.* [28], and Pan *et al.* [29] are available to the public at the time of the writing of this paper.

Dataset	n (No. of qubits)	m (No. of cycles)	M (Sample size)	F_{XEB}
Sycamore [27]	12, 14, . . . , 50, 51	14	$5 \times 10^5 \sim 2.5 \times 10^6$	
	53	12, 14, 16, 18, 20	$5 \times 10^5 \sim 3 \times 10^6$	2.24×10^{-3} for $n = 53, m = 20$
Zuchongzhi 2.0 [10]	15, 18, . . . , 54, 56	10	$10^5 \sim 2 \times 10^5$	
	56	12,14,16,18,20	2.5×10^7	6.62×10^{-4} for $n = 60, m = 20$
Zuchongzhi 2.1 [11]	15, 18, . . . , 57, 60	10		
	60	12,16,20,22,24	7×10^7	3.66×10^{-4} for $n = 60, m = 24$
Kalachev *et al.* sampling [28]	53	12, 14, 16, 18, 20	10^6	
spoofing	53	20	3×10^6	6.00×10^{-3} for $n = 53, m = 16$
Pan *et al.* spoofing [29]	53	20	$2^{20} \approx 10^6$	3.70×10^{-3} for $n = 53, m = 20$

B. Statistics of samples of random circuits

1. Datasets of random circuit sampling

We compare the three datasets of bit strings sampled from the Sycamore random circuits and the one dataset of classical random bit strings. Table I shows the datasets of the random circuit sampling where the Sycamore dataset, Kalachev *et al.*'s dataset, and Pan *et al.*'s dataset are open to the public. Note that our previous work [25] compared the Zuchongzhi dataset for $n = 53$ [10] and the Sycamore dataset. The first is the Google dataset used in quantum supremacy experiments with the Sycamore quantum processor [27]. To compare with other datasets, only the subdataset for $n = 53$ qubits and the numbers of cycles $m = 12, 14, 16, 18, 20$ are considered. Each data file is identified using five parameters: n is the number of qubits, m the number of cycles, s the seed for the pseudorandom number generator, e the number of elided gates, and p the sequence of coupler activation patterns. For example, measurement-n53-m20-s0-e0-pABCDCDAB.txt stands for the measurement data file for 53 qubits, 20 cycles and the ABCDCDAB activation pattern.

The second dataset is Kalachev *et al.*'s [17]. They produced one million bit strings for the Sycamore random circuit for up to 20 cycles using the multitensor contraction algorithms and a modified frugal rejection sampling method. To produce k bit strings, they first calculated $2k$ random batches and then selected k bit strings out of $2k$ batches using the frugal rejection sampling. We obtained Kalachev *et al.*'s data from [28]. The dataset is composed of five text files of one million bit strings for the Sycamore random circuit with $m = 12, 14, 16, 18$, and 20 cycles (for example, samples-m20-f0-002.txt). For cycle $m = 12, 14, 16, 18$, the target fidelity is 0.02 and for $m = 20$, the target fidelity is 0.02. Also, one text file for spoofing the linear XEB (spoofing-m20.txt) is provided.

The third dataset is generated by Pan *et al.* [29] using the tensor network simulation. However, only the spoofing dataset with about one million bit strings, samples-metropolis.txt, is available. Finally, we prepare the file of bit strings sampled from a classical uniform random distribution (n50-classical.txt).

2. NIST random number tests

If one takes a close look at bit strings sampled from a random quantum circuit, bit 0 and 1 seem to appear randomly. Note that we focus on the randomness of bits of a sample, not bit strings. As discussed, the probability of finding a bit string x is given by $p(x) = |\langle x|U|0^{\otimes n}\rangle|^2$ and is not uniformly random. So some bit strings are more likely to be sampled. While it is not clear whether a random circuit produces bit 0 or 1 uniformly at random, our previous studies [25] showed that some of the Zuchongzhi samples pass the NIST random number tests [40]. Recently, Shi *et al.* [41] demonstrated a possibility of using the Boson sampling method to generate true random bits. They showed that bit strings generated from the Boson sampling on optical qubits pass the NIST random number tests while more simple ways of generating true quantum random numbers exist [42]. Here, we perform the NIST random number tests [40,43] on bit strings sampled form the Sycamore random circuit using tensor network simulation: Kalachev *et al.*'s five samples and Pan *et al.*'s spoofing sample. As expected, Pan *et al.*'s spoofing sample does not pass the NIST random number tests because the first several bits of 56 bit strings are fixed to 0s or 1s. We find that all five samples of Kalachev *et al.* pass the NIST random number tests. The details of the NIST random number tests for the six samples are shown in the Supplemental Material [44]. However, we leave an open question as to whether bits of bit strings sampled from a random circuit should be uniformly at random.

3. Heat maps of bit strings

A heat map is a good way of visualizing the statistical properties of the outcomes of each qubit. The averages of the outcomes at each qubit arranged in the two-dimensional array of the Sycamore processor are plotted for the classical random bits in Fig. 1(a), the Sycamore sample in Fig. 1(b), the Kalachev sample in Fig. 1(c), and the Pan *et al.*'s sample in Fig. 1(d); Kalachev *et al.*'s sample and Pan *et al.*'s sample, both of which were obtained using the tensor network simulation, look like only that a random quantum circuit acts on only some parts of 53 qubits. Specifically Pan *et al.*'s spoofing sample is quite different from the other three samples. As the animations in the Supplemental Material show [44], some qubits of Pan *et al.'s* spoofing sample are turned always off (and turned on later).

To see the statistical difference among the samples in detail, we slice a $10^6 \times 53$ bit array, $\mathcal{D} = (x_1, x_2, \ldots, x_M)^t$, into 53×53 square arrays $\{A^{(i)}\}$ with $i = 1, \ldots, L$ and $L \equiv \mathrm{int}(M/53)$. The heat maps of the averages of $A^{(i)}$, $(1/L)\sum_i A^{(i)}$, for the classical random bit sample, the

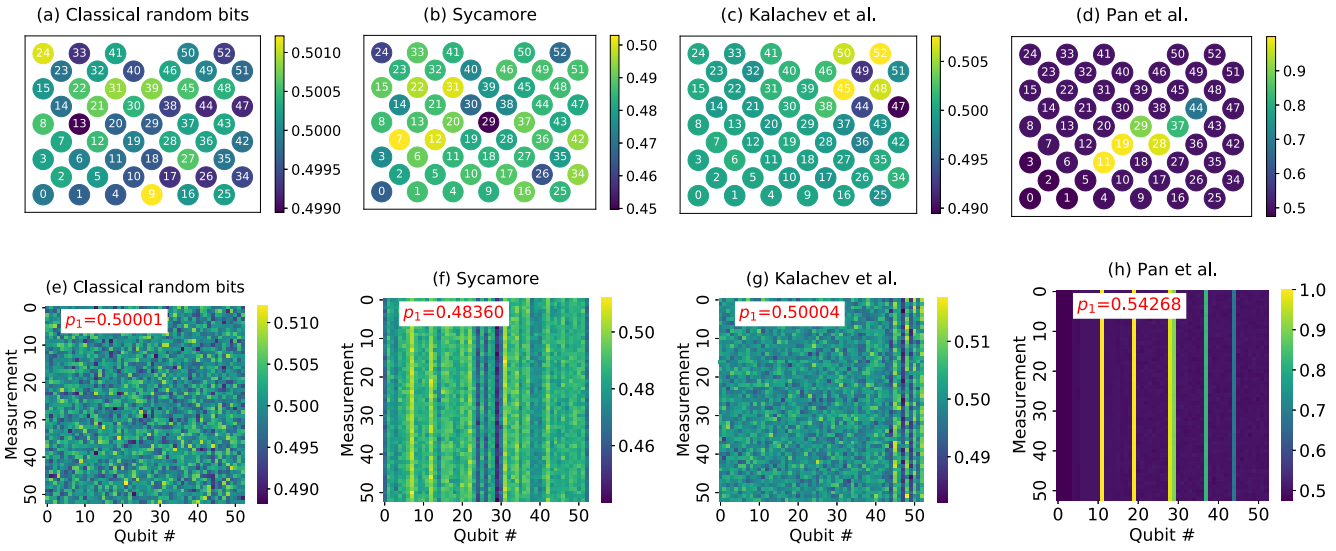

FIG. 1. (First row) For two-dimensional arrays of the Sycamore quantum processor with 53 qubits, the heat maps plot the averages of finding outcome 1 of each qubit over one million measurements for (a) the classical random bit strings, (b) the Sycamore sample (measurement-n53-m20-s0-e0-pABCDCDAB.txt), (c) Kalachev *et al.* sample (samples-m20-f0-002.txt), and (d) Pan sample (sample-metropolis.txt). The number on a circle is a qubit index ranging from 0 to 52, which is the reverse of the qubit index of the Sycamore processor. (Second row) The heat maps are plotted for (e) classical random bits, (f) the Sycamore sample, (f) Kalachev *et al.*'s sample, and (g) Pan *et al.*'s sample. The average of finding bit 1 over the sample is denoted by p_1.

Sycamore sample, the Kalachev *et al.*' sample, and the Pan *et al.*'s sample are plotted in Figs. 1(e), 1(f), 1(g), and 1(h), respectively. The heat map of the classical random bits has no stripe patterns, as expected. The heat map of Pan *et al.* sample exhibits also the stripe patterns. However, it is quite different from that of the Sycamore sample or Kalachev *et al.*'s. It is interesting to see the heat map of Kalachev *et al.*'s samples, Fig. 1(g), shows stripe patterns similar to that of the Sycamore sample, Fig. 1(f), while the first passes the NIST random number tests. One may expect that bit strings which pass the NIST random number tests would do not show any patterns. However, Kalachev *et al.*'s samples here and Zuchongzhi samples as shown in our previous work [25] may break a common belief on random bit or random numbers.

We count the number of outcome 1 of a sample composed of one million bit strings, denoted by p_1. As shown in Figs. 1(e), 1(f), 1(g), and 1(h), the averages of finding outcome 1 are estimated as $p_1 = 0.50001$ for the classical random bit sample, $p_1 = 0.48360$ for the Sycamore sample, $p_1 = 0.50004$ for Kalachev *et al.*'s sample, and $p_1 = 0.545000$ for Pan *et al.*'s sample. Note that the averages of finding outcome 1 for Zuchongzhi samples are almost 0.5. the Sycamore samples have p_1 less than 0.5 because of the readout errors [45].

4. Marchenko-Pastur distances and Wasserstein distances

One expects that two classical processors will produce the same output or statistically similar outputs. Similarly, two quantum processors implementing similar quantum gates will produce similar outcomes [46]. Suppose a unitary operator V of an ideal processor is approximate to a unitary operators U of a real processor. The error between U and V is defined by $E(U,V) \equiv \max_{|\psi\rangle} ||(U-V)|\psi\rangle||$. If the difference in two gates is small, then the measurement statistics of $U|\psi\rangle$ is approximated by $V|\psi\rangle$. This is written as the inequality [46]

$$|p_U - p_V| \leqslant 2E(U,V). \tag{3}$$

If the error $E(U,V)$ is small, the difference in probabilities between measurement outcomes p_U and p_V is also small.

If two processors, regardless of whether they are quantum or classical, implement the similar unitary operators for random circuit sampling, the statistical distances between probabilities should be close to each other. In practice, the two probability distributions, $p_U(x)$ and $p_V(y)$, may be unknown or unavailable, but only outcomes $\mathcal{D}_U = \{x_1, x_2, \ldots, x_M\}$ and $\mathcal{D}_V = \{y_1, y_2, \ldots, y_M\}$ sampled from $p_U(x)$ and $p_V(y)$ are available. Thus one needs to calculate the statistical distances between two samples $\mathcal{D}_U$ and $\mathcal{D}_V$. To this end, we employ the Marchenko-Pastur distances and the Wasserstein distances between samples.

The linear XEB, Eq. (2), is zero if the bit strings are sampled from classical uniform random bits and a sample generated from a random circuit would be different. Random matrix theory, here the Marchenko-Pastur distribution of random matrices [47], could be helpful in capturing this difference. An $M \times n$ binary array $\mathcal{D} = (x_1, x_2, \ldots, , x_M)^T$ is sliced into into the collection of $k \times n$ rectangular random matrices X. Here we set $k = 2n$. If all entries of X are independent and identically binary random variables $\{0, 1\}$, then one has the mean $\mu_X = 1/2$ and the variance $\sigma_X^2 = 1/4$. We calculate the distribution of the eigenvalues of the square matrices $(1/k)X^TX$ as shown in Fig. 2. The eigenvalue distribution of X is composed of the two parts: the bulk distribution is the Marchenko-Pastur distribution corresponding to random noise and the outlier represents the signal [24,25]. The random matrices X can be written as $Z = 2X - J$ where J has all entries 1 and Z has entries with the zero mean $\mu_Z = 0$ and the variance $\sigma_Z^2 = 1$. The square random matrix $(1/k)X^TX$ can be

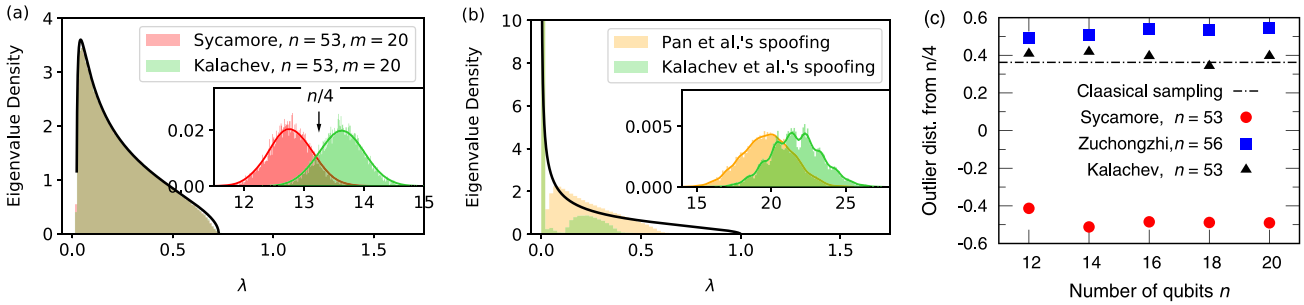

FIG. 2. (a) Marchenko-Pastur distribution of eigenvalues of $(1/k)X^TX$, Eq. (5), for the Sycamore sample with $n = 53$ and $m = 20$ (red color) and for the Kalachev *et al.*'s sample with $n = 53$ and $m = 20$ (orange color). The inset shows the outlier peaks around $n/4$. (b) Marchenko-Pastur distribution of eigenvalues of $(1/k)X^TX$ for Pan *et al.*'s spoofing sample (samples-metropolis.txt) and Kalachev *et al.*'s spoofing sample (spoofing-m20.txt). The inset shows the outliers. (c) Marchenko-Pastur distances (outlier distances from $n/4$) as a function of the number of cycles m for the Sycamore samples and Kalachev *et al.*'s samples with $n = 53$ qubits, and for the Zuchongzhi samples with $n = 56$.

expressed as

$$\frac{1}{k}X^TX = \frac{1}{4k}(Z^TZ + Z^TJ + J^TZ + J^TJ). \tag{4}$$

Here, the eigenvalue distribution of the first term in Eq. (4) is given by the Marchenko-Pastur distribution [47]

$$\rho(\lambda) = \frac{1}{2\pi\sigma^2\gamma}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\lambda}, \tag{5}$$

where $\gamma = n/k$ is the rectangular ratio and $\lambda_\pm = \sigma^2(1 \pm \sqrt{\gamma})^2$ are the upper and lower bounds. For $k = 2n$, one has $\gamma = 1/2$ and $\lambda_+ = 1 + \sqrt{1/2}$. The eigenvalues of the last term $(1/4k)J^TJ$ are 0 and $n/4$. This is the location of the outlier. We call *the Marchenko-Pastur distance* the distance of the peak position of the outlier of a sample from $n/4$. Note that, while the bulk shape of the Marchenko-Pastur distribution (5) depends on the rectangular ratio $\gamma = n/k$, i.e., the size of slices, the peak position of the outlier does not. In the Appendix, the Marchenko-Pastur distributions for the rectangular ratios $\gamma = n/k = 1/4, 1/2, 1$ are plotted. One can see that the peak positions of the outliers are not affected by the size of slices k.

Figure 2(a) plots the Marchenko-Pastur distribution of eigenvalues of the ensemble of $(1/2n)X^TX$ for the Sycamore sample (measurement-53-m20-s0-e0-pABCDCDAB.txt) and the Kalachev *et al.*'s sample (samples-m20-f0-002.txt). The majority of the eigenvalues follow the Marchenko-Pastur distribution given by Eq. (5). The inset of Fig. 2(a) shows the outliers of the distribution of eigenvalues around $n/4$, here $53/4 = 13.25$ for $n = 53$ qubits. The distance of the peak of the outliers from $n/4$, called the Marchenko-Pastur distance, could be used to measure how a sample is deviated from classical random bit strings. Figure 2(b) plots the Marchenko-Pastur distribution for Pan *et al.*'s spoofing sample (samples-metropolis.txt) and Kalachev *et al.* spoofing file (spoofing-m20.txt). As expected, the eigenvalue distributions of both spoofing samples are quite different from those of the Sycamore samples or Kalachev *et al.*'s samples. There is a big peak at eigenvalue zero, $\lambda = 0$. Also, the outliers are far away from $n/4$.

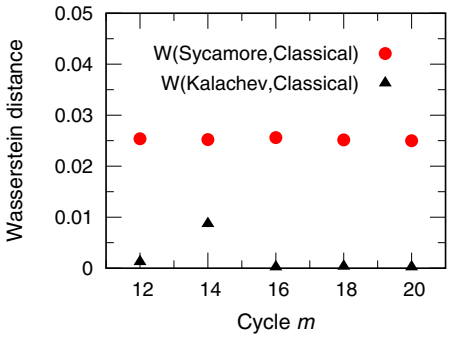

FIG. 3. Wasserstein-distances of the Sycamore samples qubits with $n = 53$ and the Kalachev *et al.*'s samples with $n = 53$ from the classical random bit sample.

Figure 2(c) plots the Marchenko-Pastur distance of samples from $n/4$ as a function of the number of cycles $m = 12, 14, 16, 18$, and 20. As shown in Fig. 2(b), the Marchenko-Pastur distances of Kalachev *et al.*'s samples are very close to that of the classical random bit strings, approximately 0.36. The closeness of Kalachev *et al.*'s samples to the classical random bit strings are consistent with the fact that Kalachev *et al.*'s samples pass the NIST random number tests discussed above. Since for Pan *et al.*'s dataset, only one spoofing sample with $n = 53$ and $m = 20$ is available, we do not plot the distance of the outliers from $n/4$ as a function of the number of cycles m.

To support the analysis above using the Marchenko-Pastur distances between samples, we calculate the Wasserstein distances for samples [48]. In contrast to the Kullback-Leibler divergence, the Wasserstein distance is a metric on probability distributions. It is symmetric and satisfies the triangular inequality. The α-Wasserstein distance between two probability distribution $p(x)$ and $q(y)$ is given by [48]

$$W(p,q) = \left(\inf_{\pi\in\Gamma(p,q)}\int_{\mathbb{R}\times\mathbb{R}}|x-y|^\alpha\, d\pi(x,y)\right)^{1/\alpha}, \tag{6}$$

where $\Gamma(p,q)$ is the set of joint distributions $\pi(x,y)$ whose marginals are $p(x)$ and $q(y)$. If $p(x)$ is the empirical distribution of a dataset $D_1 = \{x_1, x_2, \ldots, x_M\}$ and $q(y)$ is the empirical distribution of a dataset $D_2 = \{y_1, y_2, \ldots, y_M\}$, then

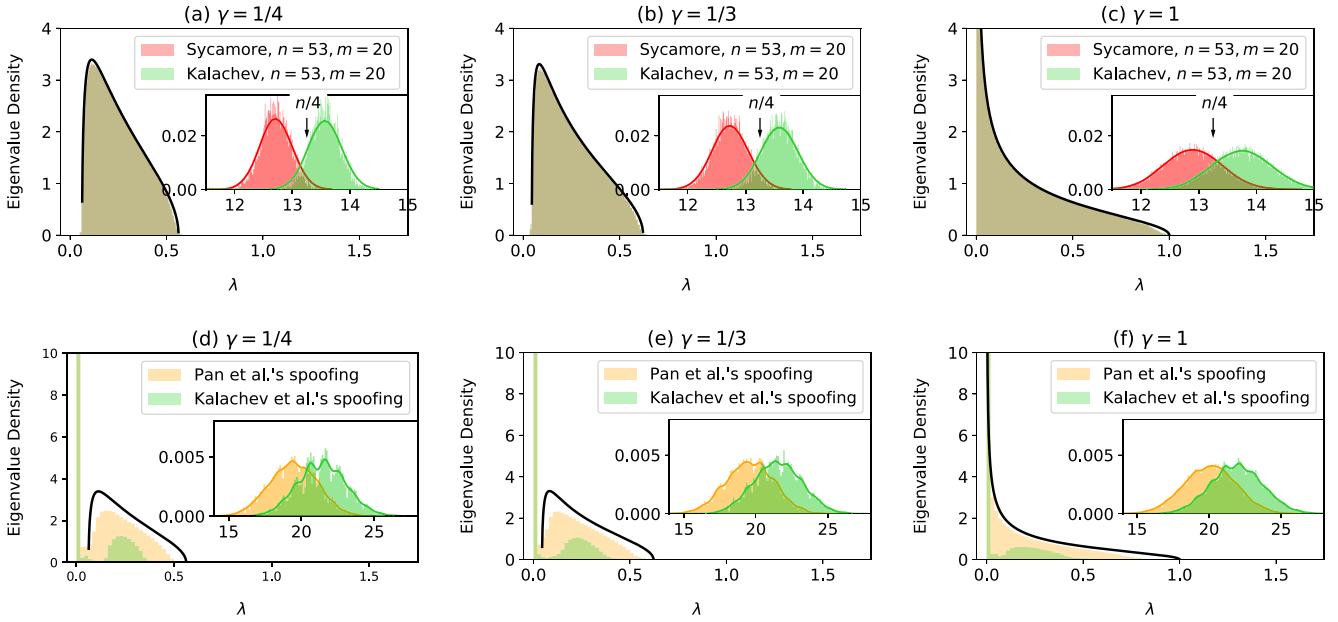

FIG. 4. The Marchenko-Pastur distributions for the Sycamore sample with $n = 53$ and $m = 20$ and Kalachev *et al.*'s sample with $n = 53$ and $m = 20$ are plotted for (a) $\gamma = 1/4$, (b) $\gamma = 1/3$, and (d) $\gamma = 1$. (d), (e), and (f) plot the Marchenko-Pastur distributions of Pan *et al.*'s spoofing sample and Kalachev *et al.*'s spoofing sample for $\gamma = 1/4$, $1/3$, and 1, respectively.

the α-Wasserstein distance is written as

$$W_\alpha(p, q) = \left(\sum_{i=1}^{M} |x_{(i)} - y_{(i)}|^\alpha \right)^{1/\alpha}, \tag{7}$$

where $x_{(k)}$ denotes the order statistic of rank k, i.e., kth smallest value in the dataset $\mathcal{D}_1$. Here, we calculate the first Wasserstein distance for the datasets of random circuit sampling using the PYTHON optimal transport library [49]. To this end, each bit string x is converted to a value in range [0,1] by dividing it by $1/2^n$. Also, we consider only one million bit strings. Figure 3 plots the Wasserstein distances of samples from the classical random bit strings as a function of the number of cycles m. It is shown that the Wasserstein distances between Kalachev *et al.*'s samples and the classical random bit strings are very close. This result is consistent with the analysis of the Marchenko-Pastur distance. The Wasserstein distances of Pan *et al.*'s spoofing sample from the classical random sample, the Sycamore sample, Kalachev *et al.*'s spoofing sample are about 0.0234, 0.01390, and 0.3275, respectively.

III. CONCLUSION

In summary, we investigated the statistical properties of bit strings sampled from random circuits on the Sycamore quantum processor and obtained using the tensor network simulation on classical processors. We considered Kalachev *et al.*'s samples [17,18,28] and Pan *et al.*'s spoofing sample [19,20,29], and compared them with the Sycamore dataset [27] and the classical random bit strings. We found that Kalachev *et al.*'s samples pass the NIST random number tests while they have stripe patterns of the heat maps. The heat maps of the two spoofing samples were shown to be quite different from those of the Sycamore samples. Since some parts of bit strings of the spoofing samples are fixed to 0s or 1s, the outcomes of the spoofing samples look like bit strings that were generated by a random circuit applied on only a few qubits. So one can easily distinguish them from the Sycamore samples while the linear XEB values of the spoofing samples are greater than 0s. Using the Marchenko-Pastur distribution of eigenvalues of the samples and the Wasserstein distances between the samples it is shown that Kalachev *et al.*'s sample is close to classical random bit strings and the spoofing samples are far away from the classical samples or from the Sycamore samples.

Google's 2019 experiment on random circuit sampling [9] was a landmark in quantum computing and has generated much interest as well as intense debate [50]. The tensor network simulation on quantum circuits has proven to be effective compared the Schrödinger-Feynman simulation and can catch up with current noisy quantum processors [17–21]. All claims are based on two metrics: computational time and the linear XEB. While the linear XEB may serve as a good measure to verify quantum advantage, its limitation was pointed out [17,20,22]. Other measures could be used to verify quantum computation for sampling [39]. For example, the random matrix analysis or the Wasserstein distances [24,25] would tell us other aspects of outcomes of random circuit sampling. As more results from quantum processors and advanced simulation on classical processors become available, a comparative study could deepen our understanding of quantum advantage. Our results raise a question about the claim that quantum advantage for random circuit sampling is faded by classical computers [20,21].

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers. We also acknowledge the National Science Foundation under Award No. 1955907.

APPENDIX: EFFET OF THE RECTANGULAR RATIO γ ON THE MARCHENKO-PASTUR DISTRIBUTION

Here, we show how the Marchenko-Pastur distribution (5) and the position of outliers are affected by the rectangular ratio $\gamma = n/k$ of a $k \times n$ matrix X, i.e., k the size of a slice of a bit-string sample. Equation (5) describes the eigenvalue distribution of the first term of Eq. (4), $(1/4k)Z^TZ$, where the mean and the variance of its entries are 0 and 1, respectively. Clearly, the shape of the Marchenko-Pastur distribution, Eq. (5), depends on the rectangular ratio γ. However, the position and the shape of the outlier are determined by the remaining three terms of Eq. (4), $(1/4k)(Z^TJ + J^TZ + J^TJ)$. Note that the eigenvalues of $(1/4k)J^TJ$ are 0 or $n/4$, and thus do not depend on γ. Figure 4 plots the eigenvalue distributions of $(1/k)X^TX$ of the Sycamore sample, Kalachev *et al.*'s, and Pan *et al.*'s for $\gamma = 1/4, 1/3, 1$. Figure 4 shows the peak positions of outliers depend little on the size of slices, k, i.e., $\gamma = n/k$.

[1] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).
[2] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).
[3] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).
[4] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Nat. Phys. **14**, 595 (2018).
[5] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya, R. Barends, B. Burkett, Y. Chen, Z. Chen, A. Fowler, B. Foxen, M. Giustina, R. Graff, E. Jeffrey, T. Huang *et al.*, Science **360**, 195 (2018).
[6] A. Bouland, B. Fefferman, C. Nirkhe, and U. Vazirani, Nat. Phys. **15**, 159 (2019).
[7] S. Scheel, arXiv:quant-ph/0406127.
[8] S. Aaronson and A. Arkhipov, *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11 (Association for Computing Machinery, New York, 2011), pp. 333–342.
[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler *et al.*, Nature (London) **574**, 505 (2019).
[10] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li *et al.*, Phys. Rev. Lett. **127**, 180501 (2021).
[11] Q. Zhu, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, S. Li *et al.*, Sci. Bull. **67**, 240 (2022).
[12] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang *et al.*, Science **370**, 1460 (2020).
[13] H.-S. Zhong, Y.-H. Deng, J. Qin, H. Wang, M.-C. Chen, L.-C. Peng, Y.-H. Luo, D. Wu, S.-Q. Gong, H. Su, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You *et al.*, Phys. Rev. Lett. **127**, 180502 (2021).
[14] C. Huang, F. Zhang, M. Newman, J. Cai, X. Gao, Z. Tian, J. Wu, H. Xu, H. Yu, B. Yuan, M. Szegedy, Y. Shi, and J. Chen, arXiv:2005.06787v1.
[15] J. Gray and S. Kourtis, Quantum **5**, 410 (2021).
[16] C. Guo, Y. Zhao, and H.-L. Huang, Phys. Rev. Lett. **126**, 070502 (2021).
[17] G. Kalachev, P. Panteleev, P. Zhou, and M.-H. Yung, arXiv:2112.15083.
[18] G. Kalachev, P. Panteleev, and M.-H. Yung, arXiv:2108.05665v2.
[19] F. Pan and P. Zhang, Phys. Rev. Lett. **128**, 030501 (2022).
[20] F. Pan, K. Chen, and P. Zhang, Phys. Rev. Lett. **129**, 090502 (2022).
[21] A. Cho, Science **377**, 563 (2022).
[22] X. Gao, M. Kalinowski, C.-N. Chou, M. D. Lukin, B. Barak, and S. Choi, arXiv:2112.01657.
[23] B. Barak, C.-N. Chou, and X. Gao, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, Leibniz International Proceedings in Informatics (LIPIcs), edited by J. R. Lee (Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2021), Vol. 185, pp. 30:1–30:20.
[24] S. Oh and S. Kais, J. Phys. Chem. Lett. **13**, 7469 (2022).
[25] S. Oh and S. Kais, Phys. Rev. A **106**, 032433 (2022).
[26] J. Eisert, D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, Nat. Rev. Phys. **2**, 382 (2020).
[27] J. M. Martinis *et al.*, Quantum supremacy using a programmable superconducting processor, Dryad, Dataset, doi: 10.5061/dryad.k6t1rj8.
[28] G. Kalachev, P. Panteleev, and M.-H. Yung, Huawei-hiq/supremacy, https://gitee.com/Huawei-HiQ/supremacy/tree/master.
[29] F. Pan, K. Chen, and P. Zhang, solve_sycamore, https://github.com/Fanerst/solve_sycamore.
[30] F. J. Dyson, J. Math. Phys. **3**, 1199 (1962).
[31] A. Hurwitz, Nachr. Ges. Wiss. Goettingen, Math. Phys. Kl. **1897**, 71 (1897).
[32] K. Zyczkowski and M. Kuś, J. Phys. A: Math. Gen. **27**, 4235 (1994).
[33] P. Diaconis and P. J. Forrester, Random Matrices: Theory Appl. **06**, 1730001 (2017).
[34] F. Mezzadri, Not. Am. Math. Soc. **54**, 592 (2007).
[35] J. Emerson, Y. S. Weinstein, M. Saraceno, S. Lloyd, and D. G. Cory, Science **302**, 2098 (2003).
[36] T. A. Brody, J. Flores, J. B. French, P. A. Mello, A. Pandey, and S. S. M. Wong, Rev. Mod. Phys. **53**, 385 (1981).
[37] M. Kuś, J. Mostowski, and F. Haake, J. Phys. A: Math. Gen. **21**, L1073 (1988).

[38] F. Haake, S. Gnutzmann, and M. Kuś, *Quantum Signatures of Chaos* (Springer-Verlag, Berlin, 2010).

[39] A. Gilchrist, N. K. Langford, and M. A. Nielsen, Phys. Rev. A **71**, 062310 (2005).

[40] L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N. Heckert, and D. Banks, A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Report No. 800-22 (2010).

[41] J. Shi, T. Zhao, Y. Wang, C. Yu, Y. Lu, R. Shi, S. Zhang, and J. Wu, arXiv:2206.02292v1.

[42] M. Herrero-Collantes and J. C. Garcia-Escartin, Rev. Mod. Phys. **89**, 015004 (2017).

[43] S. K. Ang, Randomness_testsuite, https://github.com/stevenang/randomness_testsuite#readme.

[44] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevA.107.022610 for the results of the NIST random number tests for six samples. In addition, it includes the four animation files to illustrate outcomes of bits for the two-dimensional array of 53 qubits of the Sycamore processor.

[45] Y. Rinott, T. Shoham, and G. Kalai, arXiv:2008.05177v3.

[46] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. (Cambridge University Press, New York, 2011), pp. 194–195.

[47] V. A. Marchenko and L. A. Pastur, Math. Sb. (N.S.) **1**, 457 (1967).

[48] C. Villani, *Optimal Transport: Old and New*, Grundlehren der mathematischen Wissenschaften (Springer, Berlin, 2008).

[49] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard *et al.*, J. Mach. Learn. Res. **22**, 1 (2021).

[50] G. Kalai, Y. Rinott, and T. Shoham, arXiv:2210.12753v3.