nature portfolio

https://doi.org/10.1038/s41586-025-09526-6

Supplementary information

Observation of constructive interference at the edge of quantum ergodicity

In the format provided by the authors and unedited

Supplementary Information for "Observation of constructive interference at the edge of quantum ergodicity"

Google Quantum AI and collaborators

CONTENTS

I. OUTLINE

In this supplementary text, we provide further details on the experimental procedures used in the main text, solidify the theory behind quantum advantage in computing OTOC$^{(2)}$, and show additional experimental results and classical computation methods designed as part of this study.

The quantum advantage of OTOC$^{(2)}$ is rooted in three separate claims; for the reader wishing to directly validate these, we now highlight the important parts of the supplementary material to visit:

1. The claim that our experiment faithfully reproduces OTOC$^{(k)}$: see Section II F 1.

2. The claim that $\mathrm{OTOC}^{(2)}$ is beyond the capabilities of classical computers to simulate to the same precision: see Section III C.

3. The claim that the signal size of $\mathrm{OTOC}^{(k)}$ only decays as a power law in ergodic dynamics: see Section IV.

Our supplemental begins in Section II, where we give the technical details of our experiment at a level sufficient to reproduce our results. We detail the quantum processors used (Section II A), and perform a random circuit sampling (RCS) experiment on our next generation Willow hardware, setting a new benchmark for beyond-classical performance at this task. We further outline the circuits implemented for the $\mathrm{OTOC}^{(2)}$ experiment (Section II C 2) and the error mitigation methods used (Section II F 1 and II F 4).

As part of the experimental section, we also give a dataset estimating the first-order OTOC, i.e. $\mathrm{OTOC}^{(1)}$ on large systems, using the same gate ensemble as in the main text. We demonstrate the formation of a diffusive OTOC front in the looped iSWAP pattern (as used in the main text), as compared to the gate order used in standard RCS experiments which demonstrate exponentially-decaying $\mathrm{OTOC}^{(1)}$ and $\mathrm{OTOC}^{(2)}$ signals. This allows us to demonstrate sensitivity to circuit parameters in experiments using up to 105 qubits; the full Willow chip. However, as we will explain in later sections of the supplemental, a large part of this sensitivity is not due to quantum interference, and can be accurately estimated using a Monte Carlo approximation in the Pauli-Liouville representation of the OTOC. This allows classical simulation to compete with quantum experiments in terms of accuracy. On the other hand, we find that despite this, no current implementation of classical algorithms can reproduce a set of $\mathrm{OTOC}^{(1)}$ data at a system size of 95 qubits, indicating that simulating $\mathrm{OTOC}^{(1)}$ remains a practically difficult task. Lastly, we demonstrate that even in the future event of a better classical algorithm that succeeds in approximating $\mathrm{OTOC}^{(1)}$ at large scale, we may still be able to combine classical simulation and quantum simulation to attain better accuracy than either alone, therefore maintaining quantum advantage through such a hybrid approach. However, due to the strong competition of classical methods for simulating $\mathrm{OTOC}^{(1)}$, and their potential to be further improved, we have focused on $\mathrm{OTOC}^{(2)}$ throughout discussions in the main text.

In Section III, we outline multiple classical algorithms with which we have attempted to simulate the experimental $\mathrm{OTOC}^{(1)}$ and $\mathrm{OTOC}^{(2)}$ results. We summarize the set of classical algorithms used in Table SI. The algorithms used can be roughly binned into exact algorithms with an exponential classical cost (typically in the system size), and heuristic algorithms with a polynomial classical cost but potentially unbounded error bars. We find a range of performance of these methods across $\mathrm{OTOC}^{(1)}$ and $\mathrm{OTOC}^{(2)}$ simulations, which is difficult to summarize. Nonetheless, as an indication that a classical method has had some success with OTOC simulation, we state whether our implementation was able to achieve an SNR greater than 1 for $\mathrm{OTOC}^{(1)}$ and/or $\mathrm{OTOC}^{(2)}$.

For ease of presentation, we first study the performance of our various classical techniques on simulating $\mathrm{OTOC}^{(1)}$ circuits in Section III B. Algorithms known in the literature based on matrix product state (MPS) (Section III C 7) or neural quantum state (NQS) techniques (Section III C 8), as well as algorithms based on Pauli pathing truncation (Section III C 6) or Clifford-based expansions (Section III B 4) that work well for expectation value estimation, struggle to estimate even $\mathrm{OTOC}^{(1)}$. However, as part of this work, we have developed a set of algorithms based on Monte-Carlo sampling of Pauli paths. In Section III B we outline the "vanilla" Monte Carlo algorithm which is surprisingly accurate for the ensemble of random circuits considered in this work, despite its simplicity. Then, in Section III B 2 and Section III B 3, we outline two extensions that can be tuned to arbitrarily high accuracy (at potentially exponential cost). Neither of these algorithms has been able to improve over the error-mitigated device performance of the quantum chip presented in Section II E. However, we believe that the gap is sufficiently small that further improvements might allow for classical simulation of this data. That a Markovian approximation should work so well for operator spreading is a direct consequence of our use of random circuits, which have a scrambling effect, preventing the formation of large loops in a path-integral formalism.

In Section III C, we extend the above approaches to attempt a classical calculation of $\mathrm{OTOC}^{(2)}$ for our 65-qubit circuit. In Section III C 1 and III C 2, we show that our extension to $\mathrm{OTOC}^{(2)}$ breaks the Monte Carlo approaches that worked so successfully for $\mathrm{OTOC}^{(1)}$. In lieu of good poly-time heuristics, we instead consider the cost of tensor network contraction in Section III C 4, which yield the classical estimates stated in the main text. As a possible heuristic improvement on this, in Section III C 5

Method	Section	References	Scaling	$\mathrm{SNR}_{\mathrm{OTOC}} > 1$	$\mathrm{SNR}_{\mathrm{OTOC}^2} > 1$
Exact tensor network contraction	III A, III C 4	[1, 2]	Exponential	✓	✓
Tensor network with gate removal	III C 5	-	Exponential	✓	✓
Neural quantum state methods	III C 8	[3–5]	Exponential	✗	✗
Matrix product state methods	III C 7	[6–8]	Exponential	✗	✗
Clifford-based expansions	III B 4	[1, 9]	Exponential	✓	-
			Polynomial	✗	-
Pauli pathing truncation	III C 6	[10, 11]	Exponential	✗	✗
Vanilla Monte Carlo	III C 1	-	Polynomial	✗	✗
Cached Monte Carlo	III B 2, III C 2	-	Polynomial	✓	✗
Tensor-network Monte Carlo	III B 3, III C 1	-	Polynomial	✓	✗

TABLE SI. **Table of classical computation methods tested in this work.** We link to the different sections where each method can be found, as well as any previous literature upon which our implementations are based, and whether the method (as used in our study) scales exponentially or polynomially in the system size. We finally note whether we were able to achieve an SNR > 1 for $\mathrm{OTOC}^{(1)}$ and $\mathrm{OTOC}^{(2)}$ systems (when tested), as a rough measure of whether we observe any correlation between a given approximation and the target OTOC. Note that the performance metric mentioned here does not correspond to competing with our beyond-classical claims, but on the other hand the performance of any of the above methods can potentially be further improved. For details on actual performance see the referenced sections.

we consider reducing the cost of tensor network contraction by removing gates from the circuit. This reduces the tensor network contraction to a heuristic as the error from gate removal is possible to estimate without access to the true results. In order to make any reasonable attempt at estimating the accuracy, we develop an *ad-hoc* method using Monte-Carlo simulation of the lower-order $\mathrm{OTOC}^{(2)}$ to estimate the SNR loss from removing gates. Within the uncertainty of such an estimate, this highly heuristic protocol for $\mathrm{OTOC}^{(2)}$ simulation breaks roughly even with the quantum processor in terms of time to solution. However, as the accuracy of this method has very little guarantee, we retain our claim that our $\mathrm{OTOC}^{(2)}$ simulation is beyond-classical.

We finish the supplemental by theoretically justifying our findings of large fluctuations in OTOC circuits, and giving further evidence that the hardness encountered for estimating $\mathrm{OTOC}^{(2)}$ circuits is likely broader than the methods considered here. In Section IV we develop two physical models for OTOCs in random circuits that demonstrate inverse-polynomially sized fluctuations for the values of k used in this text. The first model (Section IV A) analyses $\mathrm{OTOC}^{(k)}$ in one-dimensional spin chains with Haar-random gates, demonstrating polynomial fluctuations via perturbation theory in the inverse of the local Hilbert space dimension d, and via tensor network calculations for $d = 2$. The second model (Section IV B) is an exactly-solvable random-matrix theory in which one can find the spectrum of the correlation operator $B(t)M$ analytically. We show that the spectrum undergoes a gap closing transition as the evolution becomes more and more scrambling. We also verify numerically that a similar transition occurs in local random circuits. On general grounds, we expect that the existence of the phase transition leads to strong circuit-to-circuit fluctuations of $\mathrm{OTOC}^{(k)}$. This validates part 3) of our beyond-classical claim. Finally, in Section III C 2 we demonstrate evidence for a significant sign problem in the estimation of even the mean $\mathrm{OTOC}^{(2)}$ in Haar-random circuits. This adds to the validation of part 2) of our beyond-classical claim that was already demonstrated in Section III C. The size of the sign problem found contrasts with a similar calculation of the variance of $\mathrm{OTOC}^{(1)}$; a sign problem was noted previously in Ref. [1], but we demonstrate in Section IV A that this is not as large, as evidenced by our success in calculating circuit-specific $\mathrm{OTOC}^{(1)}$ in Section III B.

II. EXPERIMENTAL DETAILS AND ADDITIONAL DATA

A. Quantum processor details

The quantum processor used in this work consists of 105 frequency-tunable superconducting transmon qubits, connected by tunable couplers. Two of the couplers were inoperable and consequently, our experiment is conducted with a two-dimensional grid of 103 qubits.

The transmon qubits used in this work are similar in design to our recent work on quantum error correction [12]. The main difference between the two processors is the shaping of ground planes near the qubits, which resulted in improved T_1 (median = 106 μs). Other qubit-related metrics, such as maximum allowed frequencies ($f_{\max}$), frequencies at idle positions of the qubits (f_{10}), qubit anharmonicities and qubit T_2 measured using Carr-Purcell-Meiboom-Gill (CPMG) pulse sequences, are shown as integrated histograms in Fig. S1.

The median single-qubit gate fidelity is measured to be 99.95% using Clifford randomized benchmarking. The median two-qubit (iSWAP-like) gate fidelity is 99.85%, obtained using cross-entropy benchmarking (XEB). The median readout fidelity is 99.5%. The single-qubit gate, two-qubit gate, and readout times are 30 ns, 29 ns, and 600 ns respectively. All metrics are measured with all qubits in simultaneous operation, and the exact distribution of errors across the quantum processor are plotted in Fig. S2.

The iSWAP-like gates are calibrated by bringing the frequencies of two qubits into resonance for a time $t_p = 21$ ns, during which the coupler is ramped down in frequency to turn on a coupling with maximum amplitude $g_{\max}/2\pi \approx 18$ MHz. These parameters are chosen to synchronize the leakage and swap error channels [13]. Additionally, effects of distortions in flux pulses are minimized by placing nearest-neighbor qubits close in frequency and allowing additional paddings between flux and microwave pulses [2]. Lastly, we use Floquet calibration techniques to characterize the conditional phase of the iSWAP-like gates [14], which arises from the dispersive interaction between the $|11\rangle$ and $|02\rangle$ states and has a median of 0.35 rad across the processor.

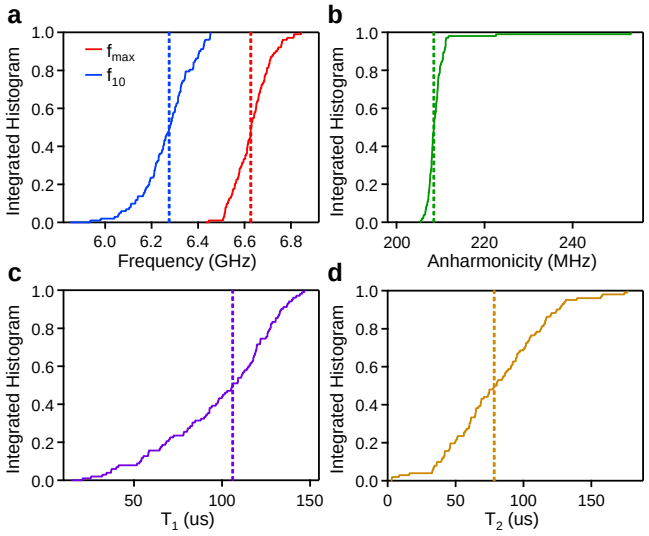


FIG. S1. **Quantum processor details.** Integrated histograms of maximum qubit frequencies $f_{\max}$ (panel **a**), qubit frequencies at idle f_{10} (panel **a**), qubit anharmonicities (panel **b**), qubit T_1 (panel **c**) and qubit T_2 (panel **d**).

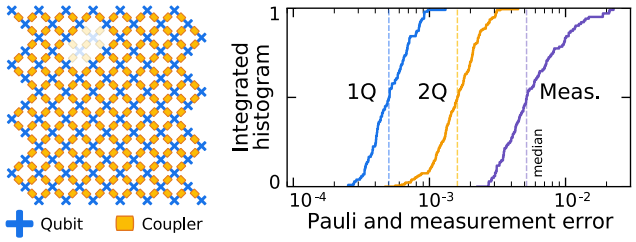


FIG. S2. **Gate and measurement errors.** Left: Layout of the 103-qubit system used for this work. Semi-transparent symbols denote broken qubits and couplers in the original 105-qubit quantum processor. Right: Integrated histogram of simultaneous operation errors.

B. RCS quantum processor benchmark

As a system-wide benchmark, we perform random circuit sampling (RCS) experiments on our quantum processor. Calculating the XEB fidelity of the full 103-qubit processor is computationally intractable for classical computers. To estimate this fidelity, we employ a "patched" approach: we modify the random circuits by removing a small subset of two-qubit gates. This dissection effectively isolates sections of qubits, creating disjoint patches for which XEB fidelity can be efficiently calculated. In our experiments (Fig. S3a), we divide our 103-qubit circuit into 3 and 4 patches (see inset of Fig. S3a), and estimate the overall circuit fidelity by multiplying the XEB fidelities of its constituent patches. To obtain each XEB fidelity data point, we averaged the results from 200 different random circuits, each run with 50,000 measurements. The estimated fidelity across different circuit depths shows close agreement with predictions from our digital error model, which includes single- and two-qubit gate and measurement errors. We estimate the overall fidelity of 103-qubit RCS circuits to be 0.1% at 40 circuit cycles, which doubles the circuit volume compared to previous records [2], and have collected 100 million sample bitstrings for a single circuit at this depth.

To quantify the computational power of our processor,

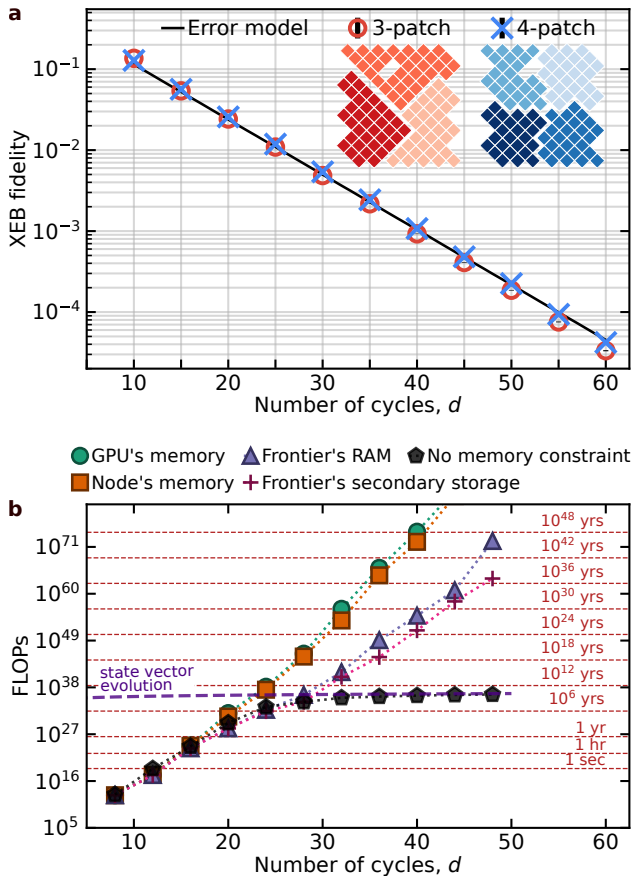


FIG. S3. **RCS quantum processor benchmark. a**, Cross-entropy bench-marking (XEB) of the 103-qubit quantum processor used in this work. The qubit grid is divided into three (red points) and four (blue points) disjoint simulatable patches which are measured in parallel (upper inset).Data correspond to the product of fidelities of the constituent patches, and solid line corresponds to digital error model prediction. **b**, Simulation time for the computation of a single amplitude of the output of the 103-qubit RCS task at different depths. We first optimize the contraction path and slicing procedure for different memory constraints imposed on the contraction. We then estimate the time it would take to contract such a tensor network assuming 20% FLOP efficiency on the Frontier supercomputer.

we estimate the classical resources required to perform a comparable RCS task. Such comparisons have become a standard benchmark in the literature [2, 15–18]. We consider Frontier, the world's most powerful supercomputer, with a peak performance of 1.71 ExaFLOPs.

Highly optimized tensor network contraction (TNC) algorithms are widely considered the most competitive classical approach for simulating near-term RCS experiments [2, 19–23] (see also Section III). These methods involve mapping the quantum circuit to a tensor network, which is then contracted to compute probability amplitudes of specific output bitstrings. This enables the sampling of bitstrings through rejection sampling [24].

The computational cost of TNC is highly sensitive to the chosen contraction ordering. To minimize this cost, we have implemented a highly efficient optimizer of contraction paths that considers techniques such as slicing to enforce memory constraints in the TNC [2, 20–23, 25] and caching and reuse of intermediate tensors [2, 21–23, 26]. Further details on these techniques and implementation of the optimizer are provided in Section III A.

In Fig. S3b, we present estimated times for computing a single probability amplitude on Frontier, a key step in simulating RCS. We analyze the scaling with circuit depth d under various memory constraints, parameterized by the maximum allowed size 2^W (where the exponent W is usually called the *width*) of intermediate tensors generated during the contraction. These constraints range from fitting within a single GPU ($W = 31$), a single node ($W = 33$), utilizing all of Frontier's RAM ($W = 48$), utilizing all of Frontier's secondary storage ($W = 54$), to even an idealized scenario with no memory constraints ($W = \infty$).

Consistent with previous studies [2, 18], we observe a crossover from exponential to linear scaling with depth in the absence of memory constraints. However, imposing memory limits enforces an exponential scaling, with tighter constraints leading to dramatically increased computational costs. Our estimates assume a 20% FLOP efficiency, which is certainly optimistic for scenarios with loose memory constraints, where bandwidth limitations become significant.

Remarkably, even in the idealized scenario without memory constraints and assuming 20% FLOP efficiency, the estimated time to compute a single output probability amplitude is about a billion years at depths $d \geq 32$. This establishes RCS on our quantum processor as a task that vastly exceeds the capabilities of current classical supercomputers.

C. Measurement circuit schematics

1. *Time-ordered correlator*

To measure the time-ordered correlator (TOC) shown in Fig. 2c of the main text, we employ the circuit schematic in Fig. S4. We initialize the system in an

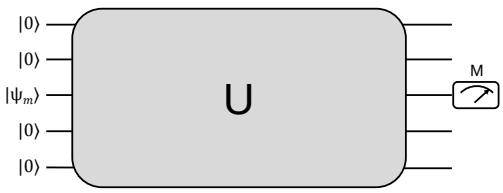


FIG. S4. **Circuit diagram for measuring a time-ordered correlator.** Circuit diagram for measuring a time-ordered correlator, $\langle M(t)M\rangle$.

eigenstate of the measurement operator M, $|\psi_{\rm m}\rangle$, then time-evolve the system with the scrambling unitary U. In the end, we measure $\langle M\rangle$ at the measurement qubit $q_{\rm m}$. Averaging over the two eigenstates of M yields a measurement outcome equivalent to $\langle M(t)M\rangle$. Similar to OTOC$^{(k)}$, we have chosen M to be Z so $|\psi_{\rm m}\rangle$ is either $|0\rangle$ or $|1\rangle$.

2. *OTOC$^{(k)}$*

The key requirement for implementing OTOC and OTOC$^{(2)}$ measurements is inverting a random circuit U. While SQ gates are readily invertible by changing the phases of the microwave drive, the iSWAP-like gates are not as straightforward due to the conditional phase, the sign of which cannot be changed with virtual Z gates alone. In our past works, this was achieved by increasing the length of the iSWAP-like gates, which reduces the conditional phase proportionally [1]. However, this approach leads to higher incoherent errors in the iSWAP-like gates and lower fidelities. The inversion also remains imperfect due to the residual conditional phase.

In our current work, we find that the iSWAP-like gates can be inverted by the following transformation: $G_{12}^{\dagger} = X_2 G_{12} X_1 Z_2^{\varphi/\pi}$. Here G_{12} is the iSWAP-like gate and φ is its conditional phase, X_1 and X_2 are Pauli-X gates applied to qubits 1 and 2, and Z_2 is a Pauli-Z gate applied to qubit 2. The additional x-rotations introduced through this transformation are then combined with the existing random SQ gates in each circuit cycle into a single layer of single-qubit gates. As such, the iSWAP-like gates are exactly inverted without any overhead in SQ gate count.

When compiling the quantum circuits for OTOC and OTOC$^{(2)}$, we remove all gates that are outside either the lightcone of $q_{\rm m}$ or the lightcone of $q_{\rm b}$. This is because they either do not contribute to the observables or have limited impact. The resulting circuit structure for OTOC$^{(2)}$ is shown in Fig. S5 (the circuit structure for OTOC$^{(1)}$ is effectively half the OTOC$^{(2)}$ circuit). To reduce idle errors, we apply the XY4 dynamical decoupling sequence [27] to qubits outside the lightcones.

Lastly, when measuring the fluctuation of OTOC$^{(k)}$ in Fig. 2c of the main text, we have adjusted the location of $q_{\rm b}$ as circuit cycle increases in order to keep the measurement close to the wavefront of $q_{\rm m}$ (i.e. the point where circuit-averaged OTOC is $\sim$0.5). For completeness, we show the choices of $q_{\rm b}$ for different cycles in Fig. S6. The effective system size, corresponding to the number of qubits within the lightcones of $q_{\rm m}$ and $q_{\rm b}$, is also shown in the same figure for reference. We note that the decrease in the effective system size at cycles 29 and 30 is due to the $q_{\rm b}$ chosen for those two cycles having more slowly growing lightcones.

D. Scrambling circuit choice

The employment of iSWAP-like gates leads to a class of maximally scrambling circuits which exhibit fast-decaying OTOCs in one-dimensional (1D) geometries [1, 28]. This presents a practical challenge for achieving experimental advantage, since the number of two-qubit gates $N_{\rm 2D}$ may not be sufficiently large to challenge tensor networks contraction algorithms before the signal size is too small to measure. We discover that, in 2D setups, scrambling may be tuned by engineering circuit structures without changing the types of gates. Fig. S7a shows two spatial patterns over which the iSWAP-like gates may be applied in 2D circuits: In the "rapid-scrambling pattern", a sequence of two-qubit gates forms a non-self-intersecting path connecting $q_{\rm m}$ and $q_{\rm b}$ (middle panel in Fig. S7a). Here maximally scrambling behavior similar to 1D geometry is expected. In the "slow-scrambling pattern" (bottom panel in Fig. S7a), gate sequences connecting $q_{\rm m}$ and $q_{\rm b}$ form loops instead. This pattern demonstrates slower scrambling and, as we will show below, qualitatively different decay of OTOCs.

Fig. S7b and Fig. S7c show, for the two different gate patterns, the spatial-temporal dependence of OTOCs after averaging over circuits with different single-qubit gates, $\overline{\mathcal{C}}^{(2)}$. We observe qualitatively different behaviors between the two gate patterns: In the slow-scrambling pattern, the decay of $\overline{\mathcal{C}}^{(2)}$ becomes slower as the distance between $q_{\rm m}$ and $q_{\rm b}$ increases. This effect, known as wavefront broadening, arises from operator spreading

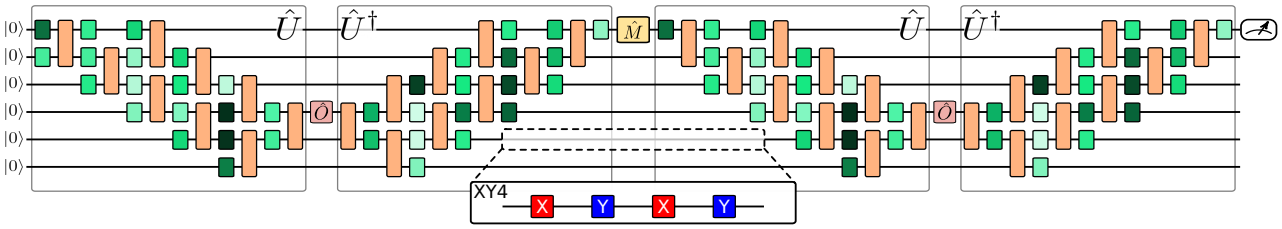


FIG. S5. **Quantum circuit for OTOC$^{(2)}$.** Detailed circuit structure of an OTOC$^{(2)}$ experiment. Quantum gates outside the lighcones of q_b and q_m are removed and replaced with dynamical decoupling (XY4) sequences.

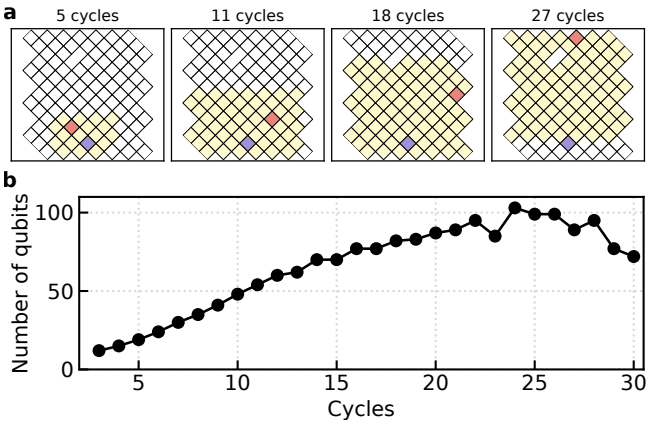


FIG. S6. **Circuit geometries for measuring OTOC$^{(k)}$ fluctuations. a,** Circuit geometries used for measuring the fluctuations of $\mathcal{C}^{(2)}$, $\mathcal{C}^{(4)}$ and $\mathcal{C}^{(4)}_{\text{off-diag}}$ in Fig. 2c of the main text. Blue square indicates location of q_m. Red square indicates location of q_b. Yellow shaded squares indicate qubits that are within the lightcones of q_m and q_b. **b,** Number of qubits within the lightcones of q_m and q_b for each cycle.

via a biased diffusion process having an average velocity slower than the lightcone [29, 30]. In the case of the rapid-scrambling pattern, the wavefront resembles a sharp step function instead. This shape is characteristic of maximally scrambling circuits, in which $B(t)$ spreads ballistically at the velocity of the lightcone [28].

Throughout the main text and the rest of this supplement, we will focus on quantum circuits that adopts the slow-scrambling pattern. We note that the diffusive behavior of the slow-scrambling circuits is in fact more representative of generic nonintegrable dynamics in 2D. For example, Haar random circuits studied in Section IV exhibit wavefront broadening similar to the slow-scrambling pattern shown here.

E. Experimental results on OTOC$^{(1)}$

In the main text, we have largely focused on OTOC$^{(2)}$ owing to the presence of large-loop interference which makes classical simulation particularly difficult. In this section, we present a complete set of experimental measurements that demonstrate two key findings: i) While the small-loop interference of OTOC allows heuristic classical algorithms to compete with experimental results at small system scales (≤ 40 qubits), the practical implementation cost of heuristic algorithms increases dramatically as system size grows. We present OTOC measurements with $\sim$100 qubits and show that without a classical supercomputer like Frontier, no practical implementation of currently known heuristic algorithms is capable of simulating such circuits. ii) Even in the event where a heuristic algorithm is capable of simulating OTOC with better accuracy than the quantum processor, the outcome of the heuristic algorithm may be combined with quantum processor results to achieve an accuracy higher than the classical simulation alone. These two findings indicate that while higher-order OTOCs are convenient choices of observables for demonstrating quantum advantage, practical tasks such as Hamiltonian could very well employ lower-order OTOCs since their simulation remains a daunting task for classical computers.

1. *Error-mitigation for OTOC$^{(1)}$*

Before presenting the experimental results related to the two findings above, we first describe the error-mitigation scheme used for OTOC$^{(1)}$ measurements. Due to the presence of noise, quantum observables such as OTOC$^{(1)}$ and OTOC$^{(2)}$ are scaled down by a noise damping factor, F_noise, that is in principle circuit-dependent. In circuits with the "rapid-scrambling" gate patterns where each Pauli string has the same butterfly velocity,

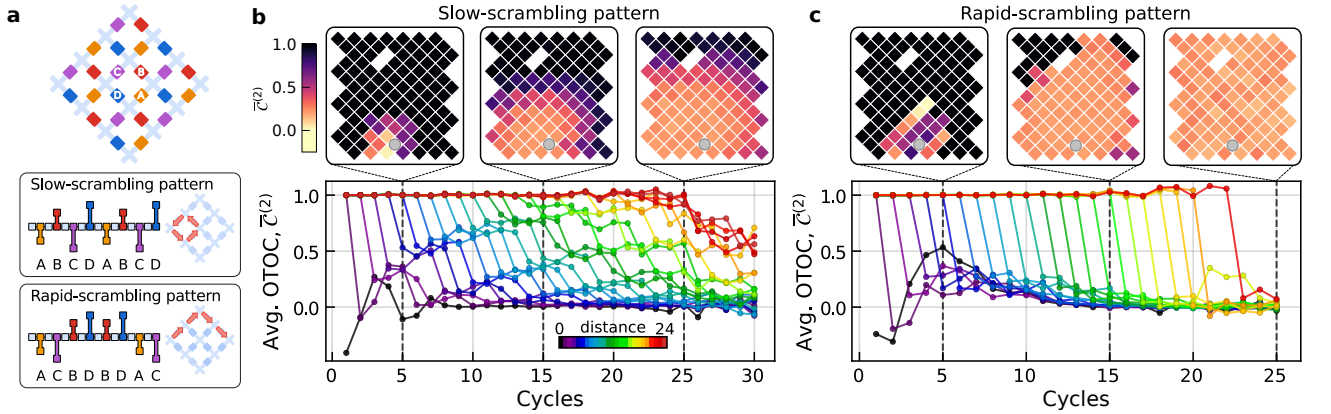


FIG. S7. **Circuit structure and decay of OTOC$^{(1)}$.** **a**, Top panel: Schematic showing two-qubit gates applied in different cycles with different colors. Bottom panels: Two circuit patterns with different orders of applying the two-qubit gates. Red arrows indicate how an imaginary "particle" is propagated in each case. **b**, Top panels: OTOC measured with different qubits as q_b and averaged over 8 circuit instances, $\overline{\mathcal{C}}^{(2)}$, for $d = 5$, 10 and 25 cycles in the slow-scrambling pattern. Grey diamond indicates the fixed location of q_m. Bottom panel: Time-dependent $\overline{\mathcal{C}}^{(2)}$ for qubits with different lightcone distances from q_m (color legend). Qubits with the same lightcone distance from q_m are averaged together. **c**, Same data as **b** but for the rapid-scrambling pattern.

the noise-damping factor on OTOC is close to the value of Loschmidt echo fidelity corresponding to $\langle M\rangle$ of q_m when the butterfly operator, B, is set to identity. This behavior was experimentally confirmed in our previous work [1].

In our current circuits which involve loops, F_noise is no longer described by the echo fidelity due to the finite spread of butterfly velocities. On the other hand, we also find that F_noise is largely insensitive to specifics of single-qubit gates and is therefore approximately a global scaling factor. To demonstrate this, we show in Fig. S8a the raw measurements of the OTOC$^{(1)}$, $\mathcal{C}^{(2)}$, for the data presented in Fig. 3d of the main text. Here it is observed that the circuit-dependent values of $\mathcal{C}^{(2)}$ from exact simulation are reproduced well in the noisy experiment but with an overall damping of amplitudes.

As a simple error-mitigation strategy, we define the accuracy metric, the signal-to-noise ratio (SNR), to be insensitive to the global scaling factor. As described in the main text, we define a set of "rescaled" OTOCs as $\mathcal{C}^{(2,s)} = (\mathcal{C}^{(2)} - \overline{\mathcal{C}}^{(2)})/\sigma(\mathcal{C}^{(2)})$, where $\overline{\mathcal{C}}^{(2)}$ and $\sigma(\mathcal{C}^{(2)})$ are respectively the mean and standard deviation of $\mathcal{C}^{(2)}$ over 50 instances. The SNR is then defined as

$$\mathrm{SNR} = \frac{1}{\sqrt{\left(\mathcal{C}^{(2,s)}_\mathrm{exp} - \mathcal{C}^{(2,s)}_\mathrm{sim}\right)^2}}, \tag{1}$$

with $\mathcal{C}^{(2,s)}_\mathrm{exp}$ and $\mathcal{C}^{(2,s)}_\mathrm{sim}$ being the OTOCs from experimental results and numerical simulations respectively. Here, subtracting the mean is motivated by the fact that

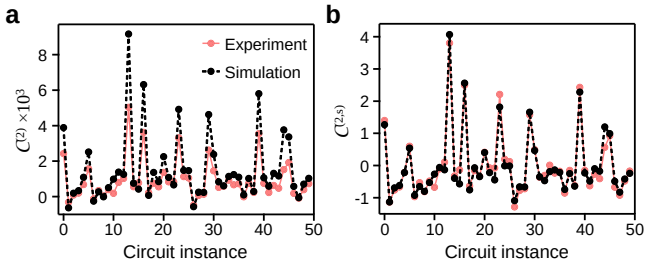


FIG. S8. **Error mitigation through global re-scaling.** **a**, Raw experimental values of $\mathcal{C}^{(2)}$ for the data shown in Fig. 3d of the main text along with values from exact simulation. **b**, Re-scaled values of $\mathcal{C}^{(2)}$, $\mathcal{C}^{(2,s)}$, for the same circuit instances and comparison with exact simulation.

circuit-averaged OTOCs are efficiently simulated through classical population dynamics. Indeed, rescaling the entire data set by $\sigma(\mathcal{C}^{(2)})$ normalizes the standard deviation to 1, thereby removing the global noise-damping factor from the signal. The rescaled OTOCs are shown in Fig. S8b, where the experiment agrees well with the exact simulation.

While our definition of SNR is insensitive to the global scale factor, we note that in practice the noise-damping factor, F_noise, can be empirically determined through comparison with efficient classical simulations using the Monte-Carlo (MC) or hybrid TNMC algorithms (described in more details in Section III). Specifically, the former is used when presenting the $\mathcal{C}^{(2)}$ data in Fig. 2c

of the main text, where the scaling factor is empirically derived from the ratio of $\sigma\left(\mathcal{C}^{(2)}\right)$ between the MC simulation and experimental Pauli-averaged OTOCs. The factor is then used to rescale the experimental OTOCs without the Pauli ensemble-averaging.

2. $OTOC^{(1)}$ experiment with 95 qubits

We now present a set of large-scale OTOC data in Fig. S9, taken with a 95-qubit geometry comprising $N_{2D} = 1000$ iSWAP-like gates. In order to compare against heuristic classical algorithms, we also need to bound the accuracy of these results similar to what was done for $\mathcal{C}^{(4)}$ in the main text. In this case, it is in fact possible to modify the quantum circuits in a way that render them classically simulatable using Monte-Carlo methods. This is done by injecting random Pauli operators everywhere in the unitary U and measure the value of $\mathcal{C}^{(2)}$ for each circuit after averaging over an ensemble of random Paulis. The averaging process effectively eliminates all quantum interference processes during the time-evolution of $B(t)$ in Pauli space, making the final amplitude of each Pauli string exactly tractable through Monte-Carlo simulation (Section III B 1).

Fig. S9 shows the experimental values of $\mathcal{C}^{(2)}$ after Pauli-averaging, which deviates significantly from the data without Pauli averaging. In particular, we find that some circuit instances (e. g. $i = 6$) are enhanced by the averaging process while other instances are suppressed by the averaging process (e. g. 46 and 48). This relative fluctuation may be understood by the fact that the interference contribution to $\mathcal{C}^{(2)}$ can have either the same or the opposite sign as $\mathcal{C}^{(2)}$, and therefore subtracting its value may lead to either suppression or enhancement of the experimental signal.

In the same figure, we have also shown results of Monte-Carlo simulation which agrees well with the Pauli-averaged experimental data up to a SNR of 3.5. Taking this as the accuracy of the non-averaged dataset, we find that our best attempt at simulating $\mathcal{C}^{(2)}$ values of these 95-qubit circuits using cached Monte-Carlo only results in SNR of $\sim$0.7 (Fig. S25b) when compared against experiment, suggesting that the simulation results likely have lower accuracy than the experimental data. Simulating each circuit in Fig. S9 using TN contraction algorithms is estimated to require ~ 3 days on the Frontier supercomputer, in comparison to ~ 30 minutes of data acquisition time on the quantum processor.

3. Turning heuristic classical simulation into error-mitigation

In this section, we demonstrate how quantum advantage may be maintained in some cases even when a heuristic classical algorithm is effective at capturing the experimental observable. Specifically, we focus on a hybrid tensor-network Monte-Carlo (TNMC) algorithm introduced in Section III B 3. This classical algorithm combines TN contraction with MC approximation along specific "cuts" (i.e. cycles) of the unitary U. The simulation outcome of this algorithm is equivalent to $\mathcal{C}^{(2)}$ after averaging over Paulis injected along the same cuts. Fig. S10a shows the results of implementing the TNMC algorithm on a set of 31 qubits, where a high SNR of 7.2 is achieved and exceeds the SNR of 6.0 achieved by the quantum processor for the same set of circuits.

Despite the better performance of the TNMC algorithm, we find that the accuracy of the quantum processor can be significantly boosted by using the TNMC results for error-mitigation. The procedure is outlined in Fig. S10: We start by computing the TNMC results along select cuts (Fig. S10a), denoted as $\mathcal{C}^{(2)}_{\text{TNMC}}$. Next, an experiment is performed which measures $\mathcal{C}^{(2)}$ with and without random Paulis inserted to the same cycles as the cut locations used in the TNMC algorithm. We denote the outcomes as $\mathcal{C}^{(2)}_{\text{exp}}$ and $\mathcal{C}^{(2)}_{\text{TNMC, exp}}$ respectively. Finally, the formula for the hybrid OTOC estimate is then:

$$\mathcal{C}^{(2)}_{\text{hybrid}} = \mathcal{C}^{(2)}_{\text{TNMC}} + \frac{\sigma(\mathcal{C}^{(2)}_{\text{TNMC}})}{\sigma(\mathcal{C}^{(2)}_{\text{TNMC, exp}})}\mathcal{C}^{(2)}_{\text{off-diag}},$$

with $\mathcal{C}^{(2)}_{\text{off-diag}} = \mathcal{C}^{(2)}_{\text{exp}} - \mathcal{C}^{(2)}_{\text{TNMC, exp}}$. The results, shown in Fig. S10b, are in good agreements with the difference between exact $\mathcal{C}^{(2)}$ and $\mathcal{C}^{(2)}_{\text{TNMC}}$. The resulting SNR, which equals 12.8 in this particular case, is now higher than the SNR = 7.2 obtained with classical TNMC simulation alone.

To demonstrate that the same observation holds irrespective of the specific implementation of the classical algorithm, we show in Fig. S11a SNRs obtained with the quantum processor alone, TNMC algorithm alone and the combination of quantum processor and TNMC algorithm. We find, for the set of 31-qubit circuits studied here, that the SNRs obtained by using both quantum and classical results outperforms the SNRs of quantum or classical results alone, regardless of the locations of the cuts. In particular, an SNR as high as 22 is obtained in the quantum-classical case for a cut placed in the middle

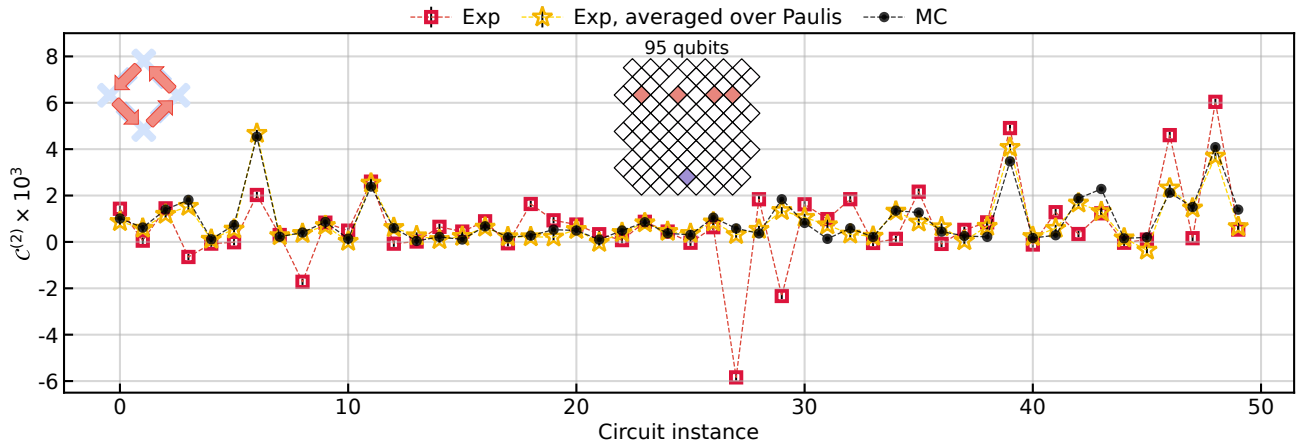


FIG. S9. **OTOC measurements with 95 qubits.** Circuit-dependent $\mathcal{C}^{(2)}$ measured on a 95-qubit geometry shown on the left (blue: $q_{\rm m}$, red: $q_{\rm b}$). The total number of two-qubit gates is $N_{\rm 2D} = 1000$. Here the error bars indicate statistical shot noise $F/\sqrt{K}$ which is much smaller than data. Experimental results after averaging over random Paulis applied everywhere in the circuits (see Fig. 3b of the main text), and comparison with Monte Carlo simulation, are also shown.

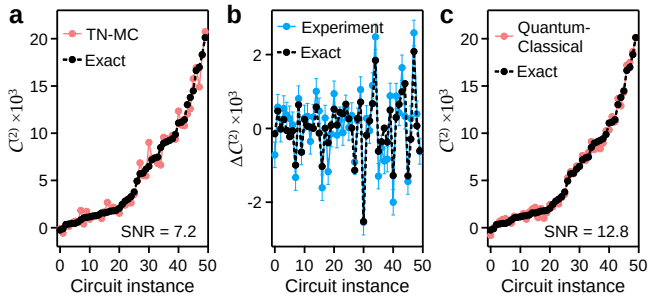


FIG. S10. **Error mitigation through efficient classical simulation. a**, Comparison of exactly simulated OTOCs and OTOCs simulated using the approximate TNMC algorithm. The circuits here include 31 qubits and a total of 24 cycles, with the "cuts" in the TNMC algorithm placed at circuit cycles 10 and 14. The circuit instances are sorted in the order of increasing OTOC values for clarity. **b**, Difference between the TNMC and exact simulation results, $\mathcal{C}^{(2)}_{\rm off-diag}$. Experimentally measured $\mathcal{C}^{(2)}_{\rm off-diag}$ values are also shown for comparison, after re-scaling by a noise-damping factor $F_{\rm noise}$ (see text). **c**, Error-mitigated $\mathcal{C}^{(2)}$ obtained by adding the experimentally measured $\mathcal{C}^{(2)}_{\rm off-diag}$ in panel **b** to the TNMC results in panel **a**. Exact simulation results are also shown for comparison.

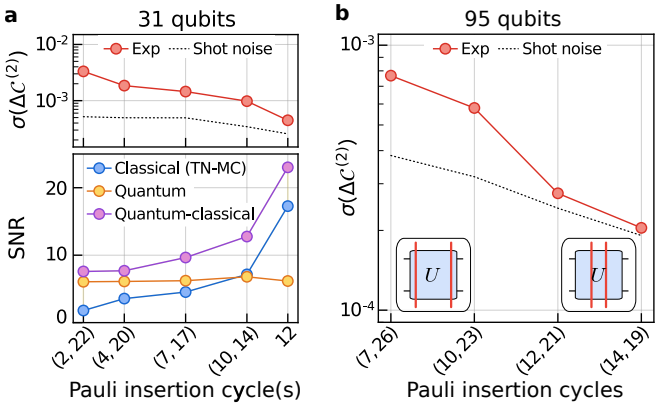


FIG. S11. **Hybrid quantum-classical OTOC measurements. a**, Top: Standard deviation of $\mathcal{C}^{(2)}_{\rm off-diag}$, $\sigma(\mathcal{C}^{(2)}_{\rm off-diag})$, over different choices of cycles for inserting random Paulis. Data are obtained with the same set of 31-qubit circuits as Fig. S10. Dashed line indicates statistical shot noise based on the number of measurements in each case. Bottom: SNR (compared against exact simulation) for the classical TN-MC algorithm with different Pauli insertion cycles and hybrid quantum-classical results at the same cycles. The SNR obtained on the quantum device alone is also shown for comparison. **b**, Similar to the top panel of **a** but measured with the 95-qubit circuits used in Fig. S9. 180 million measurement shots are taken per circuit for the (14, 19) injection cycles.

of U (cycle 12). These results indicate that as long as the difference of a classical approximation and exact quantum simulation is sufficiently large to be measured on the quantum processor above the statistical shot noise floor, quantum advantage may be maintained through the hybrid quantum-classical protocol outlined in this section.

It is instructive to apply the same protocol to the 95-qubit circuits of Fig. S9 and probe whether the difference $\mathcal{C}^{(2)}_{\rm off-diag}$ can be measured. Fig. S11b shows the experimentally measured standard deviations $\sigma(\mathcal{C}^{(2)}_{\rm off-diag})$ for

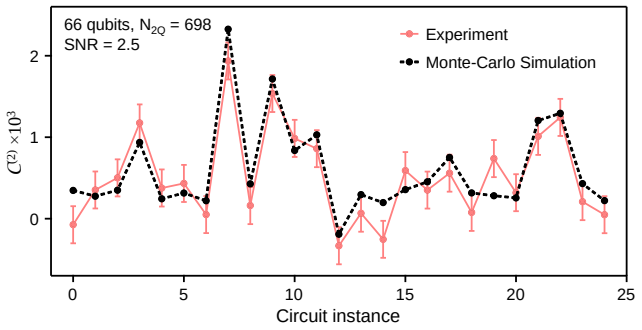


FIG. S12. **Infinite-temperature OTOC$^{(1)}$.** Experimental measurements of $C^{(2)}$ on a set of 66-qubit circuits with $N_{2D} = 698$ iSWAP-like gates. Data shown are averaged over 900 different initial states (see text for details). Monte-Carlo simulation of the same circuits is shown for comparison, which agrees with experimental results with a SNR of 2.5.

four different implementations of a two-cut TNMC algorithm. We observe that $\sigma(\mathcal{C}^{(2)}_{\mathrm{off-diag}})$ decreases as the cuts are moved toward the middle of U, which has 31 cycles in total. This is consistent with Fig. 3c of the main text. Importantly, at cut locations (12, 21) which approximately divides U into three equal parts and therefore is optimal for practical TNMC implementation, we can still resolve $\sigma(\mathcal{C}^{(2)}_{\mathrm{off-diag}})$ above the statistical shot noise floor. Based on the estimates in Section III B 3, TNMC implemented with these cuts requires the Frontier supercomputer and is therefore practically infeasible is to attempt at the moment of writing. Based on these observations and the results from the previous section, we conclude that exceeding the accuracy of the 95-qubit OTOC measurements in Fig. S9 is not possible without extensive runtimes on classical supercomputers.

4. *Infinite-temperature OTOC$^{(1)}$*

Throughout the experiments reported in the main text, we have used a product state as the initial state of the system. In real systems of interest, oftentimes the OTOC being measured corresponds to that of infinite-temperature states. In other words, $\mathcal{C}^{(2)} = \mathrm{Tr}[B(t)M]$. Here, we show that infinite-temperature OTOC can also be measured on our device.

The experimental protocol for measuring infinite-temperature OTOC consists of averaging over bitstring initial states of the form $|00101101...\rangle$ where each qubit is randomly initialized at the $|0\rangle$ or $|1\rangle$ state. When a sufficient number of initial states are used, the results are expected to converge to the infinite-temperature OTOC. To enable comparison with simulation, we fix the Rabi angles in the random single-qubit gate to be $\theta/\pi = 0.5$ rather than randomly chosen from $\{0.25, 0.5, 0.75\}$. This gate ensemble minimizes the chances of recombined trajectories in Pauli space (see Fig. 3a of the main text) based on small-scale numerical studies (not shown). As such, quantum interference effects are minimized and Monte-Carlo simulation is expected to capture the infinite-temperature OTOC relatively well.

To test this protocol, we perform initial-state-averaged measurements of $\mathcal{C}^{(2)}$ on a 66-qubit geometry with 698 iSWAP-like gates. This experiment is conducted on one of our earlier generations of quantum processors (see later discussion), which has higher gate errors compared to the current processor. Despite this, we find that the initial-state-averaged OTOCs agree well with Monte-Carlo simulation with a SNR of 2.5 (Fig. S12), which is primarily limited by statistical shot noise. The scheme outlined can be straightforwardly extended to higher-order OTOCs, such as OTOC$^{(2)}$.

F. Additional details for the OTOC$^{(2)}$ experiment

In this Section we further discuss additional details for OTOC$^{(2)}$ regarding the error-mitigation protocol used for OTOC$^{(2)}$ circuits, as well as the procedure to estimate the SNR of the 65-qubit experiment in Fig. 4a of the main text.

1. *Error-mitigation for OTOC$^{(2)}$*

Error-mitigation of OTOC$^{(2)}$ requires a different approach from OTOC$^{(1)}$. The zero-distance condition (which is satisfied by the dominant terms in $\mathcal{C}^{(4)} = \mathrm{Tr}[\rho\,(MB(t))^4]$) involves the trace of four Pauli strings, $\mathrm{Tr}[P_{\alpha_1}P_{\alpha_2}P_{\alpha_3}P_{\alpha_4}] = 1$. Unlike the case for $\mathcal{C}^{(2)}$ where the zero-distance condition only involves two Pauli strings (and, therefore, being more restrictive), two terms satisfy the zero-distance condition for $\mathcal{C}^{(4)}$: either the Pauli strings are pairwise the same (three out of the four combinations), or $\alpha_1 \neq \alpha_2 \neq \alpha_3 \neq \alpha_4$. As a result, $\mathcal{C}^{(4)}$ cannot be easily reduced to classical probabilities, rendering MC techniques ineffective.

In Fig. S13a, we show the raw OTOC$^{(2)}$ measurements for a set of 31-qubit circuits with ($\mathcal{C}^{(4)}_{\mathrm{diag}}$) and without ($\mathcal{C}^{(4)}$) averaging over random Paulis inserted in the

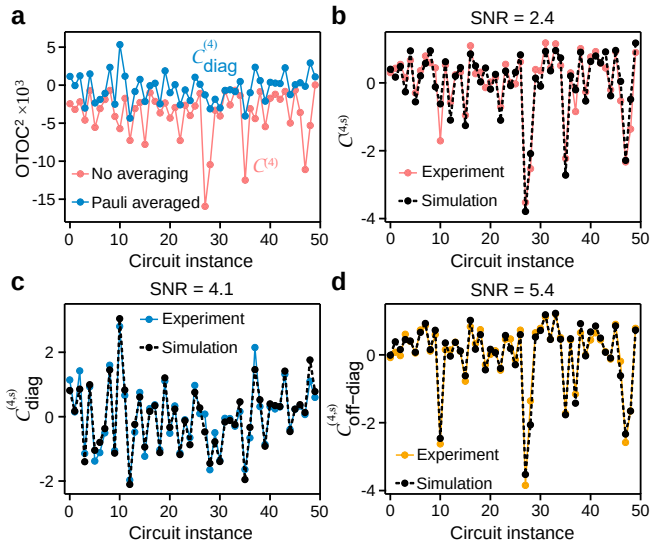


FIG. S13. **Error mitigation of OTOC$^{(2)}$.** **a**, Experimentally measured values of OTOC$^{(2)}$ with ($\mathcal{C}^{(4)}_{\text{diag}}$) and without ($\mathcal{C}^{(4)}$) averaging over random Paulis inserted into the middle of U. Data shown are without any re-scaling. **b**, $\mathcal{C}^{(4)}$ after re-scaling by its own standard deviation $\sigma(\mathcal{C}^{(4)})$, $\mathcal{C}^{(4,s)}$, and comparison with re-scaled exact simulation. **c**, $\mathcal{C}^{(4)}_{\text{diag}}$ after re-scaling by its own standard deviation $\sigma(\mathcal{C}^{(4)}_{\text{diag}})$, $\mathcal{C}^{(4,s)}_{\text{diag}}$, and comparison with simulated values of $\mathcal{C}^{(4)}_{\text{diag}}$. Here $\sigma(\mathcal{C}^{(4)}_{\text{diag}})$ is a factor of $F_{\text{noise}} = 1.6$ larger in simulation than in experiment. **d**, Re-scaled off-diagonal contribution $\mathcal{C}^{(4)}_{\text{off-diag}} = \mathcal{C}^{(4)} - \mathcal{C}^{(4)}_{\text{diag}}$, $\mathcal{C}^{(4,s)}_{\text{off-diag}} = \left(\mathcal{C}^{(4)}_{\text{off-diag}} - \overline{\mathcal{C}^{(4)}}_{\text{off-diag}}\right)/\sigma(\mathcal{C}^{(4)}_{\text{off-diag}})$, and comparison with simulation. Here $\sigma(\mathcal{C}^{(4)}_{\text{off-diag}})$ is a factor of $F_{\text{noise}} = 3.8$ larger in simulation than in experiment. In panels **b** through **d**, the mean of each data set has been subtracted, consistent with the treatment for OTOC, $\mathcal{C}^{(2)}$.

middle of U. In Fig. S13b, we compare the re-scaled $\mathcal{C}^{(4,s)} = (\mathcal{C}^{(4)} - \overline{\mathcal{C}}^{(4)})/\sigma(\mathcal{C}^{(4)})$ with numerical simulations, finding an SNR of 2.4. This modest SNR is understood to arise from the fact that noise affects the diagonal contribution $\mathcal{C}^{(4)}_{\text{diag}}$ and the off-diagonal contribution $\mathcal{C}^{(4)}_{\text{off-diag}}$ differently and rescaling by a single noise factor F_{noise} leads to inaccuracies. To test this in experiment, we measure $\mathcal{C}^{(4)}_{\text{diag}}$ explicitly through averaging Paulis inserted in the middle of circuit evolution, finding an improved SNR of 4.1 for this component after the same global rescaling procedure (Fig. S13c). Moreover, the off-diagonal contribution $\mathcal{C}^{(4)}_{\text{off-diag}}$, obtained after subtracting $\mathcal{C}^{(4)}_{\text{diag}}$ from $\mathcal{C}^{(4)}$, shows an SNR of 5.4 after global rescaling (Fig. S13d). Comparing the values of the global re-scaling factors F_{noise} obtained in both cases, we find that indeed they differ by more than a factor of 2, confirming the hypothesis above. Separating out the two contributions to $\mathcal{C}^{(4)}$ by measuring the signals after Pauli-averaging therefore serves as an effective error-mitigation strategy, beyond revealing large-loop interference as presented in the main text.

To test the robustness of the error-mitigation protocol, we conducted extensive measurements of $\mathcal{C}^{(4)}_{\text{off-diag}}$ across different system sizes. The results are shown in Fig. S14. The six sets of data conducted on Processor 1 (Willow) are used for the SNR analysis in Fig. 4b and Fig. 4c of the main text (also see next section). We find that even on earlier generation (Sycamore) quantum processors where the gate errors are nearly twice the errors on Willow, we were able to routinely obtain SNRs above 3. In particular, the 36-qubit data set on Processor 2 have a noise damping factor ($F_{\text{noise}} = 0.08$) close to the 65-qubit Willow data set in Fig. 4a of the main text ($F_{\text{noise}} = 0.05$) and still yield a SNR of 3.5.

While the noise damping factor, F_{noise}, does not affect our SNR due to its definition, it remains desirable to estimate its magnitude for OTOC$^{(2)}$ in practical applications such as Hamiltonian learning. Unlike OTOC for which efficient classical simulation such as Monte-Carlo or TNMC exist and may be used to determine the noise-damping factor F_{noise} through comparison with experimental data, no efficient classical simulation exists (to our knowledge) for $\mathcal{C}^{(4)}_{\text{off-diag}}$. Instead, we show that empirically $1/F_{\text{noise}}$ is well-correlated with the Loschmidt echo fidelity. Here the echo fidelity is defined as the average of two different values of $\langle M \rangle$ measured at q_{m} after setting either B or M to identity in Fig. S5. The results are then further averaged over all instances. As shown in Fig. S15, the echo fidelities are close to $1/F_{\text{noise}}$ over the 10 small-scale experiments shown in Fig. S14. The 65-qubit data set in Fig. 4a of the main text are therefore re-scaled by the empirically measured Loschmidt echo fidelity of 0.05. The observed correlation between $1/F_{\text{noise}}$ and Loschmidt echo fidelity is further confirmed through noisy simulations of small-scale circuits shown in the next section.

2. *Validating error-mitigation with noisy numerics*

In this section, we numerically check the error mitigation of the previous section using both simple depolarizing noise, and realistic noisy simulation including qubit decay and dephasing.

Realistic noise modeling in transmon devices typically

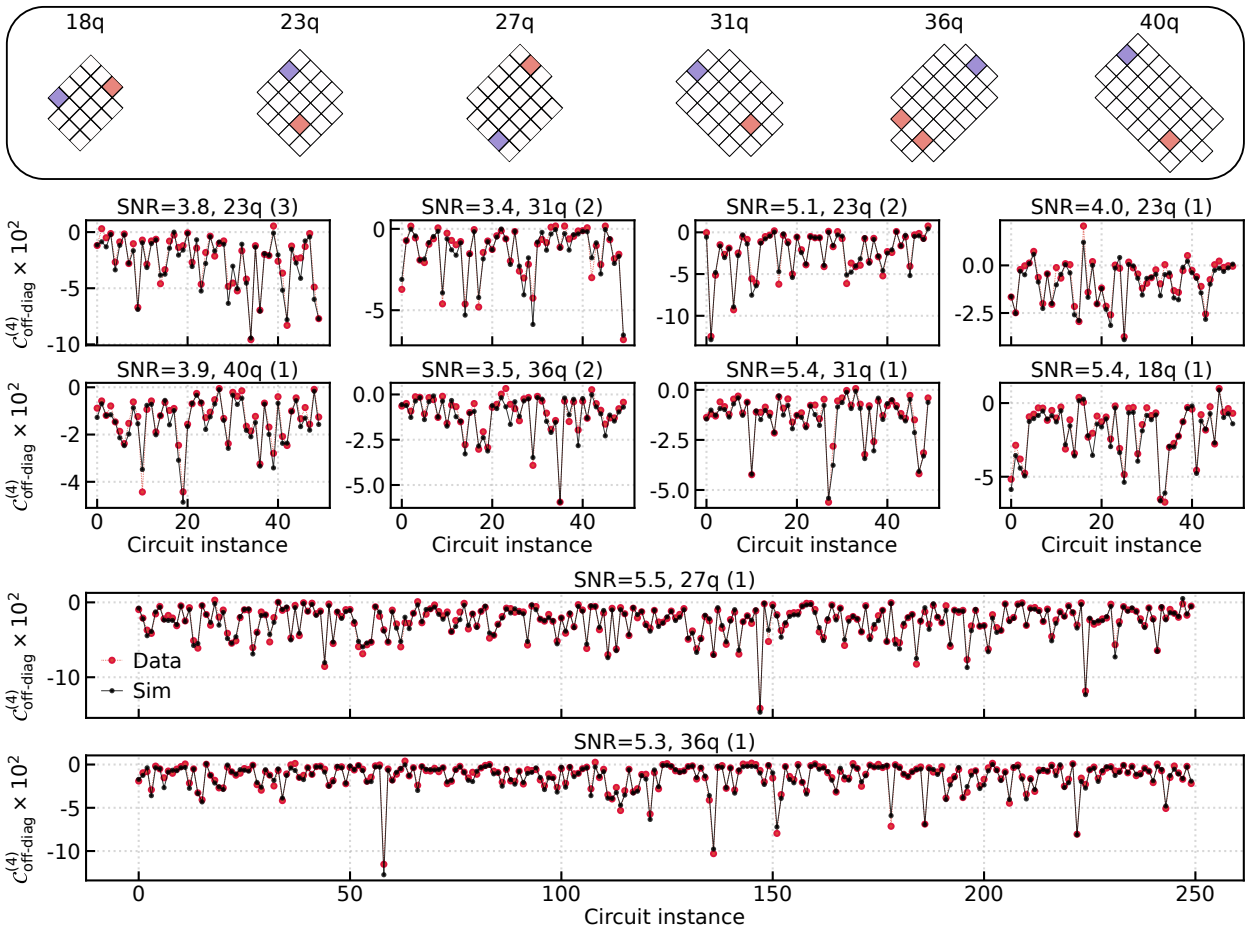


FIG. S14. **Measurements of OTOC$^{(2)}$ for different system sizes.** Experimentally measured vs exactly simulated values of $\mathcal{C}^{(4)}_{\text{off-diag}}$ for 6 different system sizes and geometries listed on the top panel. Some data are measured on older generation quantum processors with higher gate errors. The SNR, system and quantum processor used are listed as the title of each data panel. The quantum processor numbers represent the following: 1: Willow processor used in the main text. 2: Sycamore processor used in a previous work [2]. 3: Sycamore processor used in another previous work [31]. For the 27-qubit and 36-qubit geometries, we measured more circuits on Willow in order to observe the error scaling shown in Fig. 4b of the main text.

requires accounting for several physical effects, including T_1 decay, dephasing, leakage, and qubit inhomogeneity [31]. Nevertheless, we expect the signals measured in this work to be largely insensitive to the exact details of the noise model. This simplification arises from two main factors. First, in our circuit implementation, we apply randomized compiling to the two-qubit gates and dynamical decoupling pulses during periods of qubit idling; both schemes effectively transform non-Pauli error channels (e.g. coherent errors and amplitude damping) into Pauli channels. Second, by design, our circuits involve highly scrambling dynamics. This implies that local errors propagate under subsequent gates into complicated, generic operators, regardless of the initial form of the error or its exact location.

Error source	Mean	Std. dev.
T_1	80 μs	20 μs
T_ϕ	100 μs	25 μs
ϵ_{1Q}	2.6×10^{-4}	0.7×10^{-4}
ϵ_{2Q}	1.2×10^{-3}	0.4×10^{-3}
ϵ_{RO}	7.5×10^{-3}	–
$1/\Gamma_{12}$	1.5 ms	–

TABLE SII. **Estimated error rates in our device.** We report the mean and standard deviation among qubits.

To validate the above expectations, we perform noisy simulations with a realistic error model developed previously for our device [31]. The model contains the following sources of error:

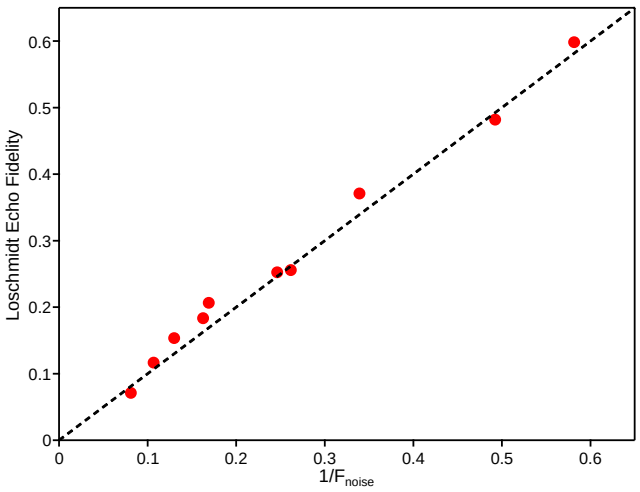


FIG. S15. **Estimating noise-damping factor from Loschmidt echo fidelities.** Loschmidt echo fidelity over the inverse noise-damping factor, $1/F_{\text{noise}}$, measured on the 10 sets of circuits shown in Fig. S14. Dashed line shows the trend corresponding to echo fidelity $= 1/F_{\text{noise}}$.

- T_1 *decay*—An amplitude damping channel (from state $|1\rangle$ to $|0\rangle$) is applied to each qubit after each circuit layer. The decay probability is $\tau/T_1(q)$, where $T_1(q)$ is the lifetime of qubit q and $\tau = 30$ ns is the time duration of each layer.

- *Dephasing*—A dephasing channel is applied to each qubit after each circuit layer. The error probability is $\tau/T_\phi(q)$, where $T_\phi(q)$ is the (white noise) dephasing time.

- *Additional MW error*—A single-qubit depolarizing channel is applied after each single-qubit MW gate. The Pauli error probability is $\epsilon_{1Q}(q)$, for a gate acting on qubit q. This accounts for additional sources of gate error, including MW crosstalk or imperfect gate calibration, not captured by $T_1(q)$ and $T_\phi(q)$.

- *Additional iSWAP error*—A two-qubit depolarizing channel is applied after each iSWAP gate, with Pauli error probability $\epsilon_{2Q}(q1, q2)$, for a gate acting on qubits q_1 and q_2.

- *Readout error*—A bit-flip channel is applied to the measured qubit immediately prior to readout, with error probability ϵ_{RO}.

- *Leakage*—A leakage transition from state $|1\rangle$ to $|2\rangle$ is applied to the each qubit after each circuit layer, with probability $\Gamma_{12}\tau$. If a leakage transition occurs, the subsequent gates acting on the qubit are removed (replaced by an identity operator), and the qubit is measured as the $|0\rangle$ or $|1\rangle$ state with equal probability. [1]

From experimental measurements, we estimate the average error rates for each noise source in our device, as well as the variation among qubits (Table SII). We model qubit inhomogeneity by sampling qubit-dependent error rates from a Gaussian distribution with the reported mean and standard deviation.

Based on this error model, we compute the $\mathcal{C}^{(4)}_{\text{off}-\text{diag}}$ for a set of 16-qubit circuits. We also compute the signal for a simplified depolarizing model that omits qubit inhomogeneity and leakage. For the latter model, we replace each of the above error channels (except leakage) with a local depolarizing channel, whose Pauli error rate is equal to the average error rate of the original channel after Pauli twirling. As shown in Fig. S16(a), the subtracted signals for the two error models are in close agreement with each other, with a relative difference of $\sim 1\%$. Moreover, after standardizing the signal across circuit instances, we find SNRs for the two noise models that differ by less than 10% [Fig. S16(b)]. In either case, an SNR larger than 10 is observed, indicating the effectiveness of the error-mitigation protocol.

From these results, we conclude that an error model consisting of uniform, local depolarizing channels is sufficient to describe the noise in our device, and thus for our subsequent analysis we adopt this simplified error model.

3. *Residual errors in error-mitigated OTOC$^{(2)}$*

In this section, we present a phenomenological model for the experimental errors that remain even after the error-mitigation protocols outlined in the previous sections. This model will be used to fit small-scale experimental data and estimate the bound on the SNR of the large 65-qubit experiment in the next section.

Notation and Concepts — For fixed t, q_{m} and q_{b}, the set of all possible circuit instances forms a probability

[1] In practice, amplitude damping, dephasing, and leakage are represented using a combined decoherence channel. For more details, see [31].

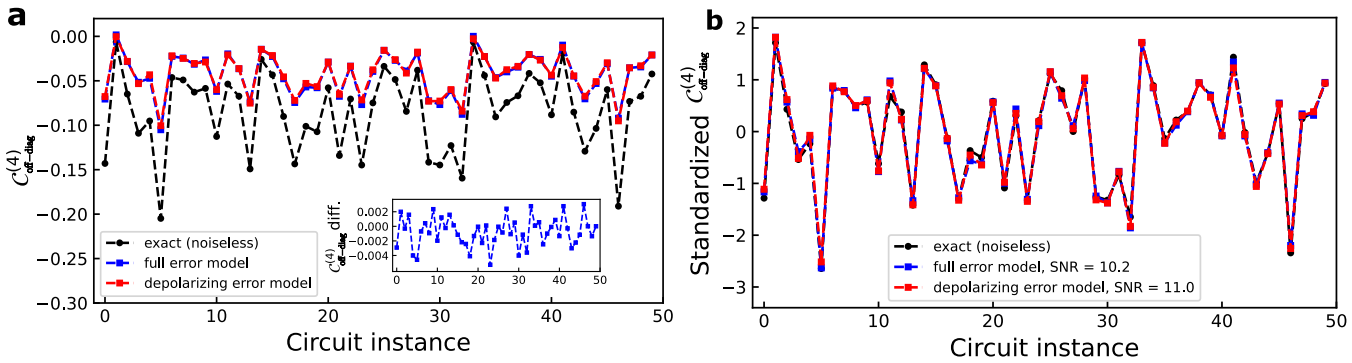


FIG. S16. **a,** Noisy simulations for $\mathcal{C}^{(4)}_{\text{off-diag}}$ in 16-qubit circuits, based on the realistic error model and a local depolarizing error model. Noiseless simulation is also shown for comparison. (inset) Difference between the two noisy simulations. **b,** Re-scaled signals used to compute the SNR for each error model.

space. From this set we draw N_{I} instances. We denote an instance average of an instance dependent variable $X(i)$ over a sample of N_{I} instances as

$$\overline{X} = \frac{1}{N_{\text{I}}} \sum_{i=1}^{N_{\text{I}}} X(i), \tag{2}$$

the variance as

$$\sigma[X]^2 = \frac{1}{N_{\text{I}}} \sum_{i=1}^{N_{\text{I}}} (X(i) - \overline{X})^2 \tag{3}$$

and the correlation with another random variable Y as

$$\rho[X,Y] = \frac{\overline{XY} - \overline{X}\,\overline{Y}}{\sigma[X]\sigma[Y]}. \tag{4}$$

In this particular section, we refer with $\mathcal{C}_{\text{nois}} \equiv \mathcal{C}^{(4)}_{\text{off-diag,noise}}$ to the difference between $\mathcal{C}^{(4)}$ and the Pauli-averaged $\mathcal{C}^{(4)}$, both measured on a quantum device affected by noise (Fig. 3 b). Similarly, $\mathcal{C}_{\text{ex}} \equiv \mathcal{C}^{(4)}_{\text{off-diag,exact}}$ denotes the corresponding result obtained from a noiseless simulation or an ideal quantum device.

Physical Mechanism Responsible for the Multiplicative Error: Fidelity Fluctuations — In the presence of noise, a measurement obtained on the noisy quantum device $\mathcal{C}_{\text{nois}}$ can be related to the exact result $\mathcal{C}_{\text{ex}}$ by introducing an instance dependent damping factor s

$$\mathcal{C}_{\text{nois}} = s\,\mathcal{C}_{\text{ex}}. \tag{5}$$

This factor s can be related to an instance dependent effective circuit volume V_{eff} [9] via the effective gate error p

$$s \sim (1-p)^{V_{\text{eff}}}. \tag{6}$$

The distribution of V_{eff} is shown in Fig. S17.

Next, we use $p \ll 1$ and assume the effective volume to be Gaussian distributed $V_{\text{eff}} \sim \mathcal{N}(\overline{V_{\text{eff}}}, \sigma_{V_{\text{eff}}})$ over the circuit instances with $\sigma_{V_{\text{eff}}} \ll 1/p$. Then s follows a log-normal distribution with mean

$$\overline{s} = (1-p)^{\overline{V_{\text{eff}}}} e^{\log(1-p)^2 \sigma^2_{V_{\text{eff}}}/2} \approx e^{-p\overline{V_{\text{eff}}}} \tag{7}$$

and variance

$$\begin{aligned} \sigma[s]^2 &= \overline{s}^2 \left(e^{\log(1-p)^2 \sigma^2_{V_{\text{eff}}}} - 1 \right) \\ &\approx (\overline{s}\, p\, \sigma_{V_{\text{eff}}})^2, &(8)\\ &\approx (\overline{s}\, \log(\overline{s})\, \sigma_{V_{\text{eff}}} / \overline{V_{\text{eff}}})^2. &(9) \end{aligned}$$

Hence, s may be considered a random variable with the above mean and variance. This shows that replacing s by its mean to estimate $\mathcal{C}_{\text{ex}}$ by $\overline{s}\,\mathcal{C}_{\text{nois}}$, we will introduce a multiplicative error of size

$$\sigma[\mathcal{C}_{\text{ex}} - \overline{s}\,\mathcal{C}_{\text{nois}}]^2 = \sigma[s]^2 \overline{C^2_{\text{ex}}}. \tag{10}$$

This perspective serves as the foundation for the error model introduced in the next section.

Noise Model for Measurements — The set of all possible circuit instances can be used to form a probability space. Therefore, the measured result $\mathcal{C}_{\text{nois}}$ and the exact result $\mathcal{C}_{\text{ex}}$ are random variables.

As discussed in the preceding section and empirically validated in Fig. 4 (b), the measurement error is observed to increase with the magnitude of the exact result, $|\mathcal{C}_{\text{ex}}|$. This motivates to model the relation between $\mathcal{C}_{\text{nois}}$ and $\mathcal{C}_{\text{ex}}$ as

$$\mathcal{C}_{\text{nois}} = s\mathcal{C}_{\text{ex}} + \epsilon, \tag{11}$$

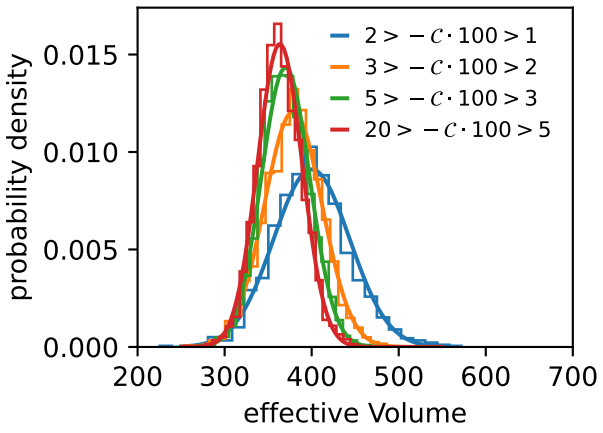


FIG. S17. **Distribution of the effective volume for $\mathcal{C} \equiv \mathcal{C}_{\mathrm{ex}}$.** Lines indicate fitted Gaussian distributions. Obtained via noisy simulation of 10000 instances of a 21 qubit-circuit with 4×68 two-qubit gates. The simulations are performed by adding in total 491 single qubit depolarizing channels after every two qubit gate with varying gate error rate p. The effective volume is obtained by fitting $\mathcal{C}_{\mathrm{nois}} = a(1-p)^{V_{\mathrm{eff}}} + c$. The average fitting error on the effective volume is about 20 for the instances with $-0.02 < \mathcal{C}_{\mathrm{ex}} < -0.01$ and about 10 otherwise.

where we introduced two more random variables: the damping factor s and the additive error ϵ. Within the model, s, ϵ and $\mathcal{C}_{\mathrm{ex}}$ are independent and follow

$$s \sim \mathcal{N}(g, g\sigma_{\mathrm{m}}), \quad \epsilon \sim \mathcal{N}(c, \sigma_{\mathrm{a}}), \tag{12}$$

where g, σ_{m}, c, and σ_{a} are model parameters to be determined.

By comparing to Eq. 9, we find them to be related to the effective volume as

$$g = e^{-p\overline{V_{\mathrm{eff}}}} \quad \text{and} \quad \sigma_{\mathrm{m}} = p\sigma_{V_{\mathrm{eff}}}. \tag{13}$$

Predictions of the Noise Model — For a dataset, which is fully described by Eq. (11) and Eq. (12), we can derive relationships between the model parameters and the statistical properties of $\mathcal{C}_{\mathrm{nois}}, \mathcal{C}_{\mathrm{ex}}$. These relationships serve to validate the model's applicability and enable predictions based on it.

First, taking the sample average of Eq. (11) yields an expression for the mean additive error

$$c = \overline{\mathcal{C}_{\mathrm{nois}}} - g\overline{\mathcal{C}_{\mathrm{ex}}}. \tag{14}$$

Next, one can relate the mean damping factor g to the correlation:

$$\rho[\mathcal{C}_{\mathrm{nois}}, \mathcal{C}_{\mathrm{ex}}] = g\frac{\sigma[\mathcal{C}_{\mathrm{ex}}]}{\sigma[\mathcal{C}_{\mathrm{nois}}]}. \tag{15}$$

From Eq. (11) we find

$$g^2\sigma[\mathcal{C}_{\mathrm{ex}}]^2 = \sigma[\mathcal{C}_{\mathrm{nois}}]^2 - \sigma_a^2 - g^2\sigma_{\mathrm{m}}^2\overline{(\mathcal{C}_{\mathrm{ex}})^2}, \tag{16}$$

which we can substitute into Eq. (15) to obtain an expression for the squared correlation

$$\rho[\mathcal{C}_{\mathrm{nois}}, \mathcal{C}_{\mathrm{ex}}]^2 = 1 - \frac{\sigma_{\mathrm{a}}^2 + g^2\sigma_{\mathrm{m}}^2\overline{(\mathcal{C}_{\mathrm{ex}})^2}}{\sigma[\mathcal{C}_{\mathrm{nois}}]^2}. \tag{17}$$

This expression can be analyzed to understand the different contributions of errors to the correlation and, therefore, the SNR $= 1/\sqrt{2(1-\rho)}$.

Maximum Likelihood Estimation and Model Validation — In this section, we will probe the applicability of the model by assessing its self-consistency. For that, we use maximum likelihood estimation (MLE) to determine the model parameters $g, c, \sigma_{\mathrm{m}}$ and σ_{a}. We then validate the model by comparing the MLE-derived parameters with the predictions of Eq. (14), Eq. (15) and Eq. (17).

To obtain an MLE estimate of the noise model parameters we need the conditional probability $p(\mathcal{C}_{\mathrm{nois}}^{(4)}|\mathcal{C}_{\mathrm{ex}}, \epsilon, s)$. It can be inferred from Eq. (11):

$$p(\mathcal{C}_{\mathrm{nois}}^{(4)}|\mathcal{C}_{\mathrm{ex}}, \epsilon, s) = \delta(s\mathcal{C}_{\mathrm{ex}} + \epsilon - \mathcal{C}_{\mathrm{nois}}) \tag{18}$$

Integrating out ϵ and s using Eq. (12), we find the likelihood function for a dataset of N measurements $\{\mathcal{C}_{\mathrm{nois}}(i)\}_{i=1}^N$ and $\{\mathcal{C}_{\mathrm{ex}}(i)\}_{i=1}^N$

$$\begin{aligned} &p(\{\mathcal{C}_{\mathrm{nois}}(i)\}_i|\{\mathcal{C}_{\mathrm{ex}}(i)\}_i) \\ &\propto \exp\sum_i \left[\frac{-\log\left(\sigma_{\mathrm{a}}^2 + [\mathcal{C}_{\mathrm{ex}}(i)]^2 g^2\sigma_{\mathrm{m}}^2\right)}{2}\right. \\ &\left. -\frac{(\mathcal{C}_{\mathrm{nois}}(i) - g\mathcal{C}_{\mathrm{ex}}(i) - c)^2}{2(\sigma_{\mathrm{a}}^2 + [\mathcal{C}_{\mathrm{ex}}(i)]^2 g^2\sigma_{\mathrm{m}}^2)}\right]. \end{aligned} \tag{19}$$

Then, the model parameters $g, c, \sigma_{\mathrm{m}}, \sigma_{\mathrm{a}}$ are set the values maximizing the likelihood function Eq. 19. The covariance matrix of the parameters is estimated using the inverse Hessian of the log-likelihood.

Model Validation — We assess the model's consistency by comparing the MLE-derived parameters with the predictions of Eq. (15) (for g), Eq. (14) (for c) and

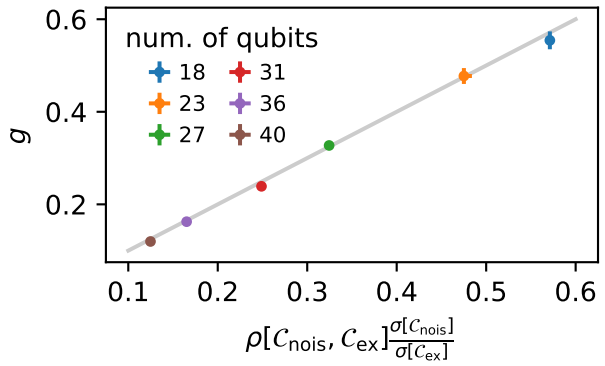


FIG. S18. **Comparison of g obtained via MLE to the model prediction $\rho[\mathcal{C}^{(4)}_{\text{off-diag}}, \mathcal{C}_{\text{ex}}]$, Eq. (15), using the experimental data for 50 instances (250 for 27 and 36 qubits).** Errors on g are estimated from the likelihood fit, errors on the sample estimate are based on the shot noise of the experimental data.

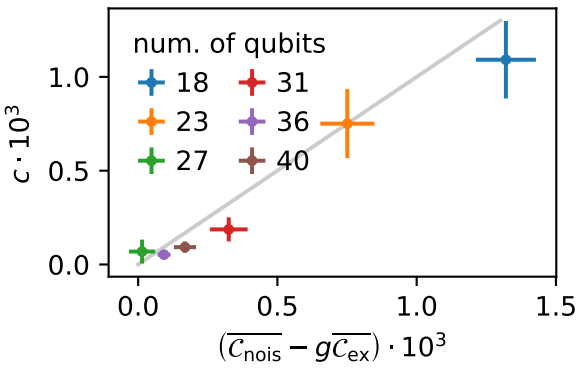


FIG. S19. **Results for c from the MLE compared to the sample estimate $\overline{\mathcal{C}_{\text{nois}}} - g\,\overline{\mathcal{C}_{\text{ex}}}$, Eq. (14) with g from Eq. (15), using the measured data for 50 instances (250 for 27 and 36 qubits).** Errors on c are estimated from the likelihood fit, errors on the sample estimate are based on the shot noise of the experimental data.

Eq. (17) (for ρ), as shown in Fig. S18, Fig. S19 and Fig. S20 respectively. We observe a good agreement between the MLE results and the model predictions: The deviation between the MLE result for g and the model prediction Eq. (15) is within the error bars, and the model prediction Eq. (14) agrees well with the MLE result for c.

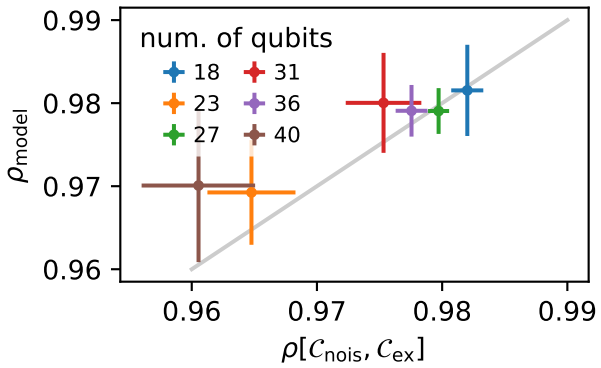


FIG. S20. **Results for the sample result $\rho[\mathcal{C}_{\text{nois}}, \mathcal{C}_{\text{ex}}]$ and $\rho_{\text{model}} = \sqrt{1 - \frac{\sigma_a^2 + g^2\sigma_m^2\overline{(\mathcal{C}_{\text{ex}})^2}}{\sigma[\mathcal{C}_{\text{nois}}]^2}}$ with the MLE-derived g, σ_m, σ_a, see Eq. (17).** We use the measured data for 50 instances (250 for 27 and 36 qubits). Errors on the model result for ρ are estimated from the likelihood fit and the shot noise of the experimental data, errors on the sample estimate $\rho[\mathcal{C}_{\text{nois}}, \mathcal{C}_{\text{ex}}]$ are based on the shot noise of the experimental data.

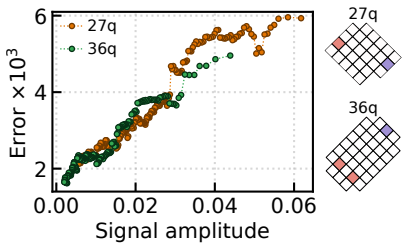


FIG. S21. **Scaling of experimental error vs signal size.** Experimental error on $\mathcal{C}^{(4)}_{\text{off-diag}}$ as a function of the signal amplitude of $\mathcal{C}^{(4)}_{\text{off-diag}}$ for two different system sizes (27 and 36 qubits, with geometries shown on the right; see Fig. S14 for raw data). Here a total of 250 circuit instances are measured for each system size and the results are sorted according to $|\mathcal{C}^{(4)}_{\text{off-diag}}|$. A rolling window analysis with a size of 50 circuits is then done to obtain the experimental error, i.e. the root-mean-square (RMS) difference between exactly simulated and experimental $\mathcal{C}^{(4)}_{\text{off-diag}}$, against signal amplitude, i.e. the RMS value of exactly simulated $\mathcal{C}^{(4)}_{\text{off-diag}}$. The number of two-qubit gates is $N_{2Q} = 284$ for 27q and $N_{2Q} = 496$ for 36q.

4. *SNR estimate for large-scale OTOC$^{(2)}$*

Having built an error model for OTOC^2, we now attempt to give an estimate on the SNR expected from the 65-qubit experiment in Fig. 4a of the main text. We first note that since exact simulation results are not available for such circuits, the distributions of error terms $s(i)$ and

$\epsilon(i)$ are not possible to characterize directly. To proceed, we make the observation that both the SNR (Fig. S14) and the scaling of errors vs signal amplitude (Fig. S21) are quite similar between the 27-qubit and 36-qubit geometry, despite a nearly factor of 2 difference in the number of two-qubit gates and noise damping factor F_{noise}. This may be related to the fact that due to similar signal sizes (i.e. $\sigma(\mathcal{C}^{(4)}_{\text{off-diag}})$) used in all of our circuit geometries, quantum volume fluctuation and therefore the distribution of errors are approximately constant over different geometries. We therefore use the multiplicative errors σ_{m} and additive errors σ_{a} obtained from fitting the 27-qubit (Willow) experimental data to infer the SNRs of all other system sizes.

Once σ_{m} and σ_{a} are fixed, the SNR of a given experimental data set may therefore be inferred by a simple simulation consisting of the following steps:

1. Scale each experimentally measured circuit-dependent value of $\mathcal{C}^{(4)}_{\text{off-diag}}(i)$ by a random multiplicative factor $s(i)$ drawn from a normal distribution with mean of 1 and standard deviation of σ_{m}.

2. Add a random value drawn from a normal distribution with mean of 0 and standard deviation of σ_{a} to the circuit-dependent values from step 1.

3. Add a random value draw from a normal distribution with mean of 0 and standard deviation of σ_{s} to the circuit-dependent values from step 2. Here σ_{s} is the statistical shot noise estimated based on the number of measurements conducted for each circuit.

4. Obtain a single SNR value after comparing the circuit-dependent values from step 3 to the original set of $\mathcal{C}^{(4)}_{\text{off-diag}}(i)$.

5. Repeat steps 1 through 4 a total of 1 million times so as to produce a distribution of SNRs. Bounds on the SNR are then obtained from the percentiles of this distribution.

Using this procedure and error parameters (σ_{m} and σ_{a}) obtained from the 27-qubit data set, we find that the experimental SNRs of the other 5 system sizes all fall within the 95% confidence interval of the SNR, as shown in Fig. 4c of the main text [2]. This agreement

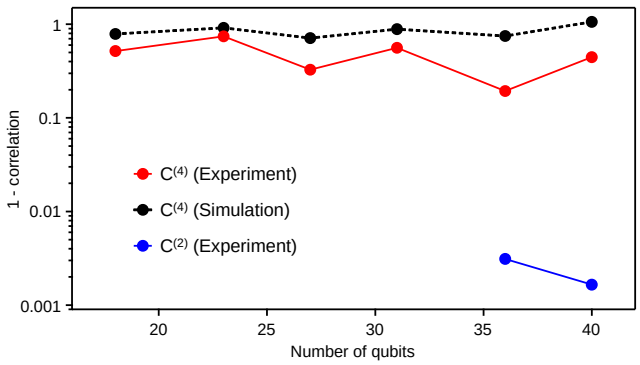


FIG. S22. **Signal change as a result of inserting Paulis into the middle of U for different system sizes.** Here correlation refers to Pearson correlation between Pauli-averaged and non-averaged signals. Results are shown for both $\mathcal{C}^{(4)}$ and $\mathcal{C}^{(2)}$. For $\mathcal{C}^{(4)}$, exact simulation results are also shown. The $\mathcal{C}^{(4)}$ circuits used here are the same as the circuits used in Fig. 4c of the main text

therefore motivates the usage of the same error parameters σ_{m} and σ_{a} to project the SNR of the 65-qubit data, which is found to have a SNR with confidence interval $[2.3, 3.5]$. Owing to potential unknown systematic errors not captured by this empirical error model, we give an estimate of 2 to 3 for the SNR.

5. *Interference effects vs system size for OTOC$^{(2)}$*

In Fig. 3c of the main text, we have contrasted the interference effects in $\mathcal{C}^{(4)}$ against $\mathcal{C}^{(2)}$ by fixing the system size at 40 and varying the location of Pauli insertion. In Fig. S22, we show similar experimental data as Fig. 4c of the main text but vary system size instead. For $\mathcal{C}^{(2)}$, it is seen that even for 31 qubits, Paulis injected in the middle of U have little impact on the observed signals, with correlation nearing 0.997. On the other hand, $\mathcal{C}^{(4)}$ manifests a much stronger signal change which barely decays as system size increases. In fact, the small decay in the case of $\mathcal{C}^{(4)}$ is not present in the noiseless simulation of the same circuits, indicating it originates from the effect of noise which reduces interference effects more at larger system sizes. This result complements the finding in Fig. 3c of

[2] The parameters σ_{m} and σ_{a} are obtained in the error model of the previous section, where errors are added to the exact values to estimate noisy values. The same parameters can be used to estimate exact values from noisy values, as long as the correlation remains high.

the main text and indicates that large-loop interference remains a significant contribution to $\mathcal{C}^{(4)}$ even at large system sizes.

III. CLASSICAL ALGORITHMS

The classical algorithms that were conceived during the course of this work are described in this section. We categorize these algorithms into two types: exact (e. g., tensor network simulation) and heuristic algorithms that attempt to approximate the true result with an uncontrolled error ϵ. For each algorithm, we describe its implementation and estimate its best-in-class performance on various trial systems. We also provide details of the analysis of the circuit cutting and gate-truncated tensor network contraction algorithms used for the time estimates in the main text.

We first comment on the choice of the algorithms we analyze. There exist numerous classical algorithms that leverage the physical properties of out-of-equilibrium quantum dynamics to reduce computational costs. Several of these methods have successfully replicated measurements of local observables [9, 32, 33]. These algorithms are effective because reproducing local observables with finite precision requires simulation of a limited effective circuit volume [9]. However, reproducing higher-order out-of-time-ordered correlators ($\mathrm{OTOC}^{(k)}$ with $k \geq 1$), where the butterfly and measurement operators are located at opposite ends of the system, requires simulating a large circuit volume that scales with the system size. This scaling suggests that algorithms optimized for local observables are likely inefficient for such tasks. Nonetheless, specialized algorithms can still reduce classical simulation costs for $k \geq 1$. In this section, we introduce classical sampling algorithms that exploit phase randomization within out-of-equilibrium dynamics to achieve these reductions.

A. Introduction to tensor network contraction

The gold standard for classical simulation of random quantum circuits is tensor network contraction (TNC). In a few words, a tensor network is a collection of multi-index arrays that are multiplied together [34, 35]. For instance, a matrix is a tensor of two indices M_{ij} and the multiplication of two matrices can be written as $\sum_j M_{ij} M'_{jk} = M''_{ik}$. When the equation involves multiple tensors and summations, the order (also called "path") in which tensors are multiplied is crucial [34, 36]: indeed, different paths can have costs in floating point operations (FLOPs) that could differ by orders of magnitude [19]. Finding optimal paths is known to be hard,[3] and exact approaches are limited to small TNCs. Significant effort has gone into developing this over recent years [2, 19–23, 25]. Advancements have not only improved contraction path optimizations, but also became aware of memory constraints [25], the advantage of caching intermediate results in memory [2, 22, 23], as well as the possibility to compute several circuit output amplitudes with a cost similar to that of a single one [2, 21–23]. These improved techniques allow for the increasingly more precise estimation of the cost for executing tensor network based random circuit sampling simulations on the world's largest supercomputer, Frontier.

Given a circuit C, the exact expectation value can be computed by converting the expectation value of $\langle Z \rangle = \langle 0| CZC^\dagger |0\rangle$ to a TNC. However, computing the exact expectation value can be demanding, as the expected FLOPS to contract a TNC tends to grow exponentially with the depth of the TNC. To limit the contraction cost, we decomposed the expectation value as a sum of orthogonal projections on all qubits at the measurement level excluding the measurement qubit and a few gates around it (up to 20 open qubits in total):

$$\begin{aligned} \langle Z \rangle &= \langle 0| CZ \left[\sum_x \Pi_x \right] C^\dagger |0\rangle \\ &= \sum_x \left[\langle 0| C\Pi_x \right] Z \left[\Pi_x C^\dagger |0\rangle \right] \\ &= \sum_x \langle \psi_x| Z |\psi_x\rangle \end{aligned} \tag{20}$$

with $|\psi_x\rangle$ a partial quantum state for a given projection x. The exact expectation value is obtained by summing over all the possible projections x. However, an approximate expectation value can be obtained by using a small number of projections [1]. Indeed, observing that $p_x = |\langle \psi_x | \psi_x \rangle|^2$ can be interpreted as probability, with $\sum_x p_x = 1$, it is possible to use rejection sampling to sample random projections x and compute approximate

[3] To be precise, finding optimal path is an NP-Hard problem [37].

expectation values as:

$$\begin{aligned}\langle Z\rangle &= \sum_x \langle\psi_x| Z |\psi_x\rangle \\ &= \sum_x p_x \frac{\langle\psi_x| Z |\psi_x\rangle}{p_x} \\ &= \mathbb{E}\left[\frac{\langle\psi_x| Z |\psi_x\rangle}{p_x}\right]. \end{aligned} \tag{21}$$

That is, the expectation value $\langle Z\rangle$ is reduced to computing the expectation value over the classical distribution induced by p_x of $\psi_x = \frac{\langle\psi_x|Z|\psi_x\rangle}{p_x}$. In general, the exact sampling from p_x is impossible because it requires computing all partial $|\psi_x\rangle$, hence the exact expectation value. Inspired by the *frugal* sampling in [24], we perform rejection sampling by exactly computing p_x for a small number of randomly chosen projections $\{x_1, x_2, \cdots, x_k\}$ from the uniform distribution, and then accepting/rejecting them with a probability $r_x = \frac{p_x}{\max_{\{x_1, x_2, \cdots, x_k\}} p_x}$. While such sampling algorithm does not guarantee that the accepted projections will have the same probability as been sampled from the exact probability distribution (if not in the limit of $k \propto 2^n$, with n being the number of qubits), we observe that it works well in practice.

The tensor network contraction optimizer (`TNCO`) — We developed a high-performance novel package, `TNCO`, for optimizing the contraction path of a tensor network and estimating its runtime (including the constraints from finite random access memory on a supercomputer). A tensor network contraction path (TNC) in `TNCO` is represented as a tree, with the root of the tree being the output tensor. If any of the intermediate tensors in the TNC cannot fit in memory, indices are *sliced* until the memory of all tensors (defined as $w = \sum_i \log_2 d_i$, with d_i being the i-th tensor index) is not larger than a given target width. Unless specified otherwise, for each Frontier's node we are assuming the maximum width of 33.[4] TNCs are updated by proposing local updates on the corresponding contraction tree, which are accepted or rejected using Metropolis-Hastings. Representing TNCs as trees allows us to limit the optimization landscape to only topologically different configurations, avoiding the extra cost to explore symmetrical contractions.

As observed in [2], optimal contractions without memory constraints may not be well suited for slicing. To slice a TNC, `TNCO` simultaneously optimizes the contraction path and the location of slices. Furthermore, `TNCO` uses hyper-indices to connect diagonal gates, and provides paths caching the intermediate tensors: if an intermediate tensor fits in memory, the subtree of such tensor is pruned from the TNC and the tensor is cached to be reused across multiple slices. To keep the cost of running `TNCO` limited, we do not account for them in the total memory count. [5]

B. Classical simulation of OTOC$^{(1)}$

In this section, we discuss classical methods used to simulate OTOC$^{(1)}$. We begin with subsections III B 1, III B 2 and III B 3, wherein we define a family of powerful *Monte Carlo* methods, which exploit having exactly two copies of the time-evolved butterfly operator.

1. *Introduction to Monte Carlo algorithms*

Let us split the time-evolution of the butterfly operator into T pieces (either individual gates or groups of those), and write

$$U = \prod_{t=0}^{T-1} g_t. \tag{22}$$

Having two copies of a time-evolved butterfly operator opens a prospect to a powerful Monte Carlo family of methods based on the *diagonal approximation*. Consider a time-evolved butterfly operator $B(T) = U^\dagger B U$ expanded in the Pauli basis $B(T) = \sum_P b_P P$, where P are all possible 4^n Pauli strings, with n being the number of qubits.

For the sake of discussion, let us temporarily consider the computation of a trace-OTOC. Plugging this Pauli decomposition into the expression for trace-OTOC, we

[4] Each node in Frontier has 8 GPUs, for a total amount of 512 GB of high-bandwidth memory.

[5] Since the total amount of required memory without caching is at most 4 times the size of the largest intermediate tensor, limiting the width is effective to ensure that the full contraction can be performed in memory. However, this is no longer true when caching is involved and the full contraction might not fit in memory at fixed maximum tensor width.

obtain

$$\mathcal{C}^{(2)} = \sum_{P,P'} b_P b_{P'} \mathrm{Tr}\left[PMP'M\right] = \sum_P |b_P|^2 (-1)^{\{P,M\}}, \quad (23)$$

where b_P are squared normalized, and $(-1)^{\{P,M\}} = \pm 1$ if P and M either commute or anti-commute.

The appearance of the positive measure $|b_P|^2$ allows one to sample the "observable" $\mathcal{O}(P) = (-1)^{\{P,M\}}$ using Monte Carlo

$$\mathcal{C}^{(2)} \approx \frac{1}{M} \sum_{P\sim|b_P|^2} \mathcal{O}(P), \quad (24)$$

where the M samples are drawn from the $|b_P|^2$ distribution. Here, $P \sim |b_P|^2$ means sampling a Pauli string P with the probability induced by $|b_P|^2$. Importantly, direct sampling from $|b_P|^2$ is hard. In the next sections, we will introduce several Monte Carlo methods, which only differ in the way they approximate the $|b_P|^2$ distribution in order to be able to sample from it. To introduce a common way of approximation, we will invoke the *path integral* formalism.

To this end, we will view the time-dependent butterfly operator $|B(t)\rangle$ as a vector in the linear operator space satisfying the condition $|B(t=0)\rangle = |B\rangle$. Action of a gate on the operator $B(t+1) = g_t^\dagger\, B(t)\, g_t$ can be viewed as an action of an induced transfer matrix G_t with elements

$$[G_t]_{pq} = \mathrm{Tr}\left[P g_t^\dagger Q g_t\right] \quad (25)$$

in the operator linear space $|B(t+1)\rangle = G_t|B(t)\rangle$.

For the sake of clarity, let us assume that B is a Pauli string such that $P_0 = B$. In this notation, an element of the distribution reads

$$\begin{aligned} |b_P|^2 &= \langle P|G_{T-1}\cdots G_0|B\rangle\langle B|G_0^\dagger\cdots G_{T-1}^\dagger|P\rangle \\ &= \sum_{\boldsymbol{Q},\boldsymbol{Q'}} \langle P|G_{T-1}|Q_{T-1}\rangle\cdots\langle Q_0|G_0|P_0\rangle \\ &\qquad \times \langle P_0|G_0^\dagger|Q_0'\rangle\cdots\langle Q_{T-1}'|G_{T-1}^\dagger|P\rangle \quad (26) \end{aligned}$$

where we have inserted an identity resolution $\sum_Q |Q\rangle\langle Q|$ after each gate with Q being a Pauli string, and a bold symbol $\boldsymbol{Q} = \{Q_0, Q_1, \ldots, Q_{T-1}\}$ denotes the collection of resolution insertions over all time stamps t.

The path integral representation Eq. (26) opens the prospect for a family of approximations, which we call diagonal approximations. For the rest of this Section, we describe multiple Monte Carlo algorithms based on the diagonal approximation.

Vanilla Monte Carlo — The first example of such approximation is the *vanilla* Monte Carlo, where the condition $\boldsymbol{Q} = \boldsymbol{Q'}$ is enforced, yielding to

$$|b_P|^2_{\text{vanilla}} = \sum_{\boldsymbol{Q}} |\langle P|G_{T-1}|Q_{T-1}\rangle|^2 \ldots |\langle Q_0|G_0|P_0\rangle|^2, \quad (27)$$

which presents a direct receipt of sampling P from $|b_P|^2_{\text{vanilla}}$: begin with $|P_0\rangle$, act a transfer matrix $G_0|P_0\rangle$, sample $Q_0 \sim |\langle Q_0|G_0|P_0\rangle|^2$, proceed to the next step for Q_1 and so on.

Observe that this approximation may be seen as ignoring all possible *cycles* (loops) in the butterfly operator evolution tree in the Pauli basis. Indeed, the information about two Pauli paths possibly rejoining later is neglected, and sampling takes place at every node of the tree.

Cached Monte Carlo (CMC) — Looking at Eq. (26), it is immediate to realize that there are $n \cdot (T-1)$ locations in *space* and *time* where $\boldsymbol{Q}$ and $\boldsymbol{Q'}$ can assume the same value, that is the locations where the two paths join. Therefore, the condition $\boldsymbol{Q} = \boldsymbol{Q'}$ can be seen as a collections of pairs of two numbers $(x,\, t)$ that enforce the constraints $Q_{x,t} = Q'_{x,t}$ for all x and t. Consequently, the vanilla Monte Carlo can be improved by relaxing as many of such constraints as possible. This leads to the idea of the *cached* Monte Carlo (CMC). Practically, we work in a sparse representation of $|B(t)\rangle$ and begin with applying the induced gates to $|B\rangle$. When this representation overfills the memory at time t, we choose some qubit x, and enforce $Q_{x,t} = Q'_{x,t}$. This enforcement creates a positive measure, and we immediately sample

$$Q_{x,t} \sim \langle B(t-1)|Q_{x,t}\rangle\langle Q_{x,t}| \otimes \mathcal{I}_{n-1}|B(t-1)\rangle\rangle, \quad (28)$$

where $|Q_{x,t}\rangle\langle Q_{x,t}| \otimes \mathcal{I}_{n-1}$ is a projector onto a Pauli operator $Q_{x,t} \in \{I, X, Y, Z\}$ at a site x and time t. That is, this sampling allows one to filter the sparse representation of $|B(t)\rangle$ by only keeping the Pauli strings which have the operator q at a site x. Having reduced the memory footprint, one can proceed with applying further gates until the procedure needs to be repeated.

Unlike the vanilla Monte Carlo, this procedure is capable of capturing *short* loops in the Pauli evolution tree, as it can act several transfer matrices sequentially

without sampling, thus correctly accounting for interferences between different Pauli paths which result in the same Pauli string.

Tensor-network Monte Carlo (TNMC) — Lastly, instead of sparsely representing the time-evolved operator $|B(t)\rangle$, one can directly sample

$$P_{t+1} \sim |\langle P_{t+1}|G_t|P_t\rangle|^2 \tag{29}$$

by transforming a slice of the circuit into a tensor network and using sampling methods for shallow circuits such as Ref. [38]. This procedure is repeated over K samples to target a desired Monte Carlo accuracy. We call this application of tensor network contraction *tensor-network* Monte Carlo (TNMC). Observe that, if $T = 1$, there is no approximation and P_N would be sampled with the exact distribution. Although, for T sufficiently large, the resulting tensor network is shallow enough to be advantageous to use tensor network methods.

Monte Carlo methods applied to state-OTOC — Let us consider a state-OTOC with a particular state $|\psi\rangle$. In such case, starting from the initial butterfly vector $|B\rangle$, we still apply the induced transfer matrices G_t and enforce diagonal constraints as in Eq. (27), (28) or (29). In the last step, however, we compute the observable differently,

$$O = \sum_{P,P'} b_P b_{P'} \langle\psi|PMP'M|\psi\rangle \tag{30}$$

for the vanilla and cached Monte Carlo. For TNMC, we instead contract a shallow tensor network to obtain $\langle\psi|(g_{T-1}Pg_{T-1}^\dagger M)^2|\psi\rangle$, where P was sampled in the last step, and g_{T-1} is the remaining part of the full unitary thereafter.

Relation to the Pauli insertion averaging — Finally, we show that a diagonal approximation of the form $Q_{x,t} = Q'_{x,t}$ is related to the Pauli insertion averaging protocol which could be done experimentally or within state-vector simulations. To this end, we split the whole unitary in two parts: before the projection at time t, U(t, 0), and after the projection $U(T, t)$. We then perform the Pauli expansion of the partially time-evolved butterfly operator $B(t) = U^\dagger(t, 0)BU(t, 0) = \sum_P b_P P$. Now, suppose that we also inserted a Pauli operator $Q_{x,t} \in \{I, X, Y, Z\}$ at a site x at time t and average over these options. The resulting expression reads

$$\mathcal{C}^{(2)}_{\text{ins.}} = \frac{1}{4} \sum_{Q_{x,t},P,P'} b_P b_{P'} \langle U^\dagger(T,t)Q_{x,t}PQ_{x,t}U(T,t) \times MU^\dagger(T,t)Q_{x,t}P'Q_{x,t}U(T,t)M\rangle, \tag{31}$$

where $\langle\ldots\rangle$ could denote either the trace or the state expectation. Importantly, the Pauli products are simplified as $Q_{x,t}PQ_{x,t} = Ps(P_x, Q_{x,t})$, where $s(P_x, Q_{x,t}) = \pm 1$, depending on whether P_x and $Q_{x,t}$ commute or anticommute. Because

$$\frac{1}{4} \sum_{Q_{x,t}} s(P_x, Q_{x,t})s(P'_x, Q_{x,t}) = \delta_{P_x,P'_x}, \tag{32}$$

we get

$$\mathcal{C}^{(2)}_{\text{ins.}} = \sum_{P,P'|P_x=P'_x} b_P b_{P'} \langle U^\dagger(T,t)PU(T,t)M \times U^\dagger(T,t)P'U(T,t)M\rangle, \tag{33}$$

which means that experimentally (or in state-vector simulations) averaging over correlated Pauli insertion affects the OTOC computation exactly in the same way, as the point-like projection in Eq. (28). Clearly, several such projections (Pauli insertion averages) can be applied at the same time, for instance in a fixed-time cut manner as done in the main text.

In the next sections, we delve into the details of implementation, performance and complexity of those variations of the Monte Carlo approach.

2. *Cached Monte Carlo (CMC): details*

The cached Monte Carlo scheme (CMC) is in principle heuristic (as we cannot bound the error), but that converges to the exact simulation in the limit of an exponentially-large cache size. This flexibility allows us to test this method on the OTOC experiments presented in the main text with available classical hardware. For a 40-qubit experiment presented in the main text, the cached Monte Carlo gives the experiment-level SNR ≈ 5.4, while for the 95-qubit experiment with two cuts we obtained an SNR above random guessing, but below the estimated SNRs of the experiment. In this section, we provide a few specific implementation details.

As discussed in Section III B 1, the cached MC algorithm stores the non-zero elements of the coefficients $\boldsymbol{b}(t)$ of the Pauli, which we call "cache" $\mathcal{C}$, of size $|\mathcal{C}|$. If the cache overfills, the method uses a point-like projection

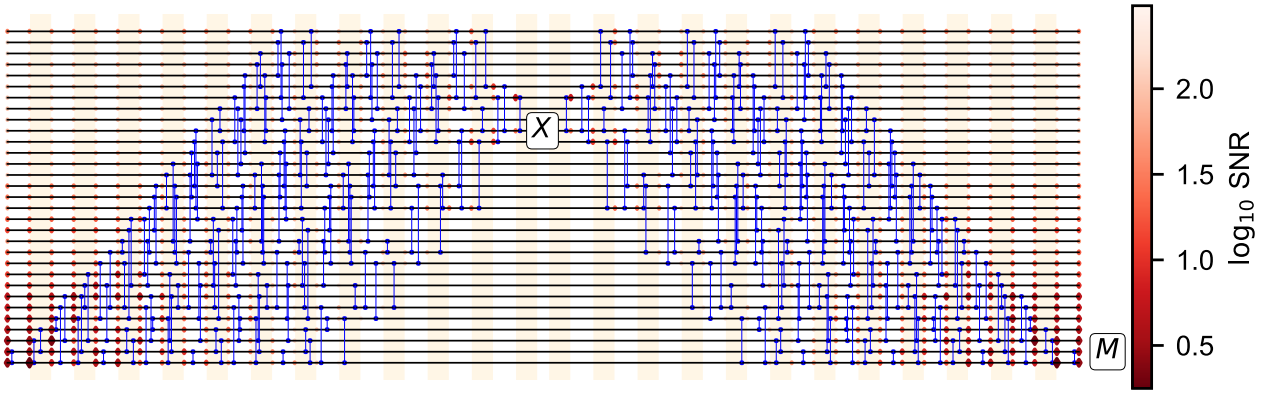


FIG. S23. **SNR of projecting in the Pauli basis.** A layout of a two-dimensional 31-qubit OTOC circuit. The shaded orange regions identify layers of two-qubit gates, and the blue lines representing two-qubit gates ate shifted horizontally within a layer for visibility. The application of a unitary U is followed by the butterfly operator X, inverse unitary $U^\dagger$ and the measurement operator. The colorbar represents the SNR reduction upon insertion of a projection at a given space-time location (q, t). The image is symmetric, since the Monte Carlo algorithm works in the Pauli space.

Eq. (28).

Computation of the expectation value — After applying all gates from U and projecting along the way, the final cache $\mathcal{C}^{\{Q_{x,t}\}}$ depends on the set of projected Paulis $\{Q_{x,t}\}$ and the projection outcomes. Most often, we compute OTOC on top of a bitstring, which is an eigenstate of the measurement operator M. In particular, let us consider $M = Z$ and $|\psi\rangle = |00\ldots\rangle$. In such case, to avoid quadratic in $|\mathcal{C}|$ computation time, we first form the resulting "wave function" as

$$|\psi_{\{Q_{x,t}\}}\rangle = \sum_{(b_P,P)\in\mathcal{C}} b_P\,(P|0\rangle)\,, \tag{34}$$

and write the OTOC as

$$\mathcal{C}^{(2)} \approx \frac{1}{M}\sum_{\{Q_{x,t}\}} \langle\psi_{\{Q_{x,t}\}}|M|\psi_{\{Q_{x,t}\}}\rangle. \tag{35}$$

where M is the number of Monte Carlo samples. The required number of samples is smaller than in the vanilla Monte Carlo, since the norm of the expectation values $\langle\psi_{\{Q_{x,t}\}}|M|\psi_{\{Q_{x,t}\}}\rangle$ decays algebraically with the cache size.

The choice of the projected qubit $Q_{x,t}$ — We have empirically observed that the impact on the simulation SNR depends on the particular choice of the projected qubit, once the cache overfills. In particular, the spatial and temporal proximity of a projection to the measurement operator is an important indicator of the systematic bias introduced by the projection. Fig. S23 illustrates the

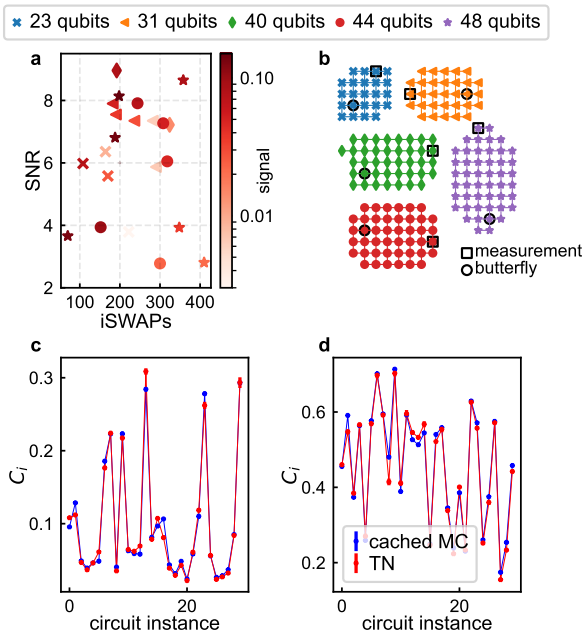


FIG. S24. **Benchmarking cached Monte Carlo on small systems.** **a,** The SNR between the cached MC and the exact simulation, **b.** The circuit geometries. For each geometry, we considered a sequence of circuit depths to vary the signal (standard deviation of the OTOC values over circuit instances). The largest exactly-contracted circuit has 410 iSWAP gates on 48 qubits at 29 cycles. **c-d.** 48-qubit comparison between the tensor-network contraction and the cached MC at depths 28 and 24, respectively.

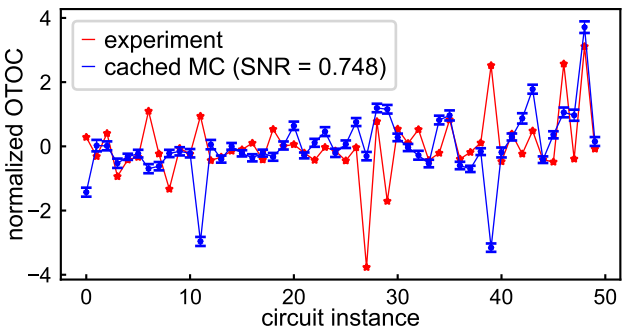


FIG. S25. **Comparison between the experiment and cached MC at 95-qubit circuits with 1000 iSWAPs.**

difference between in-bulk and idle projections, as well as the impact of proximity to the measurement operator. We observe that projections appearing closer to the measurement and butterfly operators lead to a greater reduction in SNR. We also observe that for the qubit lines far from the measurement operator, the in-bulk projections are more harmful than the idling projections.

Small Pauli path truncation — An important part of the algorithm is to truncate small wave function elements $b_P \in \mathcal{C}$. Since the vector $b_P(t)$ is normalized, we truncate the cache elements with

$$|b_P|^2 < \frac{\texttt{tol}}{|\mathcal{C}|} \tag{36}$$

and typically choose $0.001 < \texttt{tol} < 0.01$.

Results on smaller systems — Before applying this algorithm to the full chip, we first assess its capacity on smaller systems where we can compute the OTOC circuits exactly, either using the tensor network contraction or the exact state-vector simulation. For these simulations, we set $|\mathcal{C}| = 50 \times 10^6$ and $\texttt{tol} = 0.001$. We perform the exact simulations of the systems with 23, 31, 40, 44, and 48 qubits and with up to 410 iSWAP gates. For each geometry, we consider 30 random circuit instances. The system geometries and the resulting SNR as a function of the signal are shown in Fig. S24. The observed SNRs are consistently well above random guessing.

Application to the 95-qubit circuit — Finally, we apply the cached Monte Carlo algorithm to the 95-qubit $\mathrm{OTOC}^{(1)}$ experiment with 1000 iSWAP gates and 29 cycles, as reported in the main text.

In a simple model with the rescaled signals being uncorrelated and instances being drawn from the normal distribution $\mathcal{N}(0,1)$, the SNR over M circuit instances follows the inverse χ_{M-1} distribution, which yields

$$\langle \mathrm{SNR}^{\text{uncorrelated}} \rangle_{\text{samples}} = \frac{\sqrt{M}}{2} \frac{\Gamma(M-1)}{\Gamma(M-1/2)} = \tag{37}$$
$$= \frac{1}{\sqrt{2}} \left(1 + \frac{5}{4M} + \mathcal{O}(M^{-2})\right).$$

In Fig. S25b, we show the comparison between the cached MC and the experiment without any Pauli averaging. In this case, the SNR = 0.748 is close to the theoretical estimate for uncorrelated signals, $1/\sqrt{2} + 5/(4\sqrt{2}M) = 0.724$ at $M = 50$, indicating that the cached MC algorithm, limited by the memory of a single GPU, is unable to capture the correlations in this case.

The cost of a perfect simulation without projections — Here, we estimate the amount of memory required to perform this algorithm on the 95-qubit 1000-iSWAP circuit with only cache truncations of the type Eq. (36), but avoiding projections. To this end, we perform a simulation of the 95-qubit circuit with a cache size limit of $|\mathcal{C}| \leqslant 200 \times 10^6$. During the simulation, the cache becomes overfilled, and a qubit is chosen for projection to free space. We track how many Pauli strings are stored in the cache as a function of the number of non-projected qubits. We observe that the cache limit of $|\mathcal{C}| \leqslant 200 \times 10^6$ may already saturate at 15 non-projected qubits, which yields a cache size scaling of $|\mathcal{C}| \sim 3.57^{N_{\text{non-projected}}}$. Since these circuits involve at most 40 unprojected qubits, this would necessitate an impractically large cache size of $|\mathcal{C}| \leqslant 3.57^{40} \sim 10^{22}$.

3. *Tensor Network Monte Carlo (TNMC): details*

As discussed in Section III B, Pauli string P_{t+1} can be sampled by converting the expression in Eq. (29) to a tensor network. This is done by a simple extension of the method of Ref. [38] from bitstrings to Pauli strings. This sampling method has a complexity dominated by the evaluation of $\mathrm{Tr}[P_{t+1} G_t P_t]$ by tensor network contraction (TNC). In addition, we focus on state-OTOC, which involves the computation of the remaining

$$\langle \psi | (g_{T-1} P_{T-1} g_{T-1}^\dagger M)^2 | \psi \rangle \tag{38}$$

as explained in Section III B, in practice carried out via TNC. Our implementation makes use of NVIDIA's

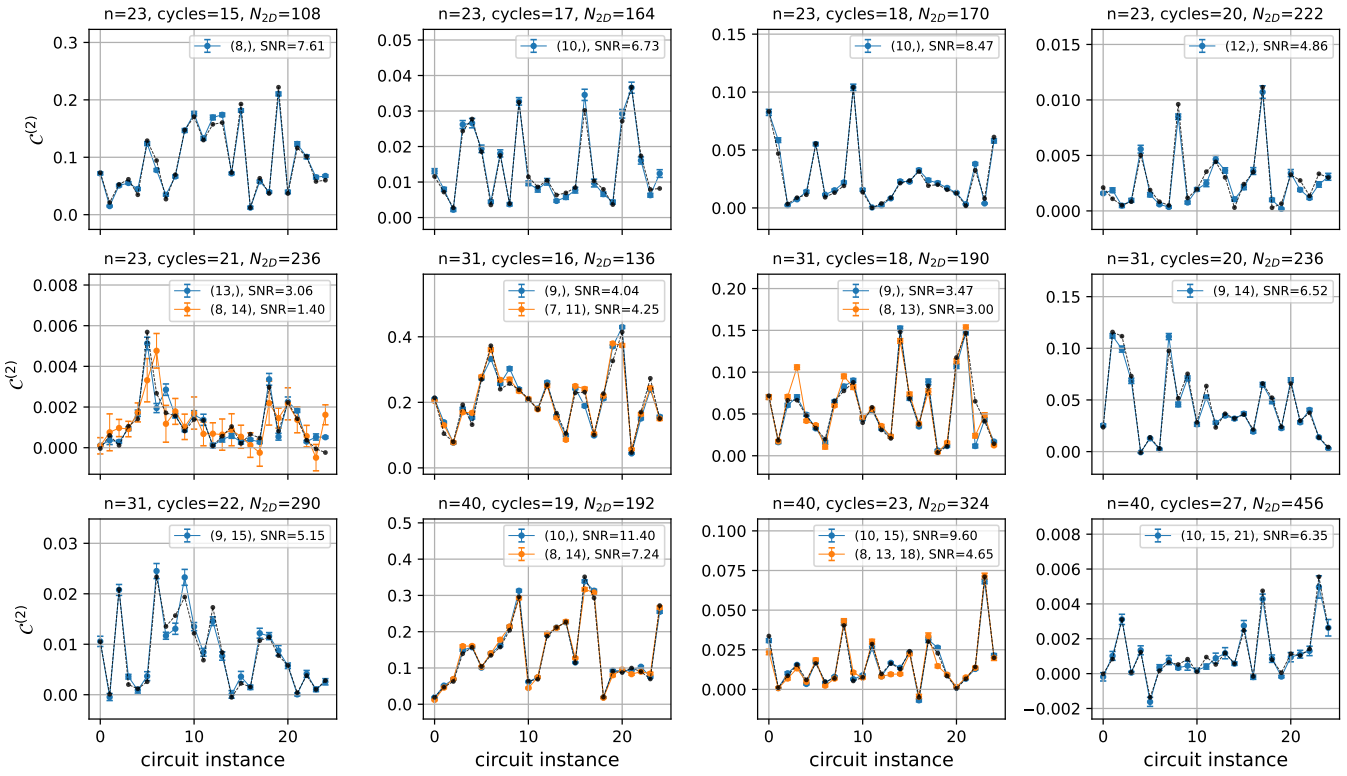


FIG. S26. **Performance of TNMC.** $\mathcal{C}^{(2)}_{\text{TNMC}}$ as a function of circuit instance for different geometries of OTOC circuits of $n = 23$, 31, and 40 qubits. The exact value of $\mathcal{C}^{(2)}$ is shown in black. The cut scheme used in each case and the SNR achieved is shown in the legend. In all cases we use $K = 5000$ Monte Carlo samples per point.

`cuQuantum SDK` library for the optimization and computation of TNC on GPUs.

Choosing circuit cuts — Given an OTOC$^{(1)}$ with unitary U, TNMC "cuts" U into T shallower circuits at

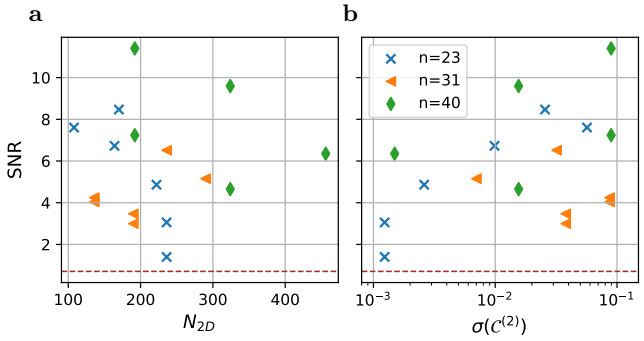


FIG. S27. **SNR achieved by TNMC for different circuit geometries.** Scatter plot of the SNR as a function of the number of iSWAP-like gates N_{2D} (**a**) and the signal size $\sigma(\mathcal{C}^{(2)})$ (**b**). We consider circuits of different number of qubits n and different depths (see Figs. S26). The baseline SNR of $1/\sqrt{2} \approx 0.71$ is shown as a dashed, brown line.

$T-1$ depth locations. Once the circuit is cut, we need to sample P_{T-1} and evaluate expression (38) a number of times. Sampling P_{T-1} involves the computation of a number of TNCs that is linear in the number of gates in the circuit. The aggregated cost of these contractions is dominated by the largest of the contraction costs of all $\text{Tr}[P_{t+1}G_tP_t]$ over subcircuits g_t, which scales exponentially in the size of G_t. For the reasons presented, it is important to choose the location of the cuts carefully, while trying to minimize the number of cuts $(T-1)$ in order to achieve a good approximation. In practice, given a certain value for T, we optimize the TNCs involved and choose the cutting scheme that minimizes computational cost. If none of the options is computationally practical, then we explore schemes with larger values of T until a viable cutting scheme is found. For large circuits this might involve a large number of cuts, and hence a poor approximation of the OTOC$^{(1)}$ values.

Performance of TNMC over small circuits — In order to study the performance of TNMC, we focus first

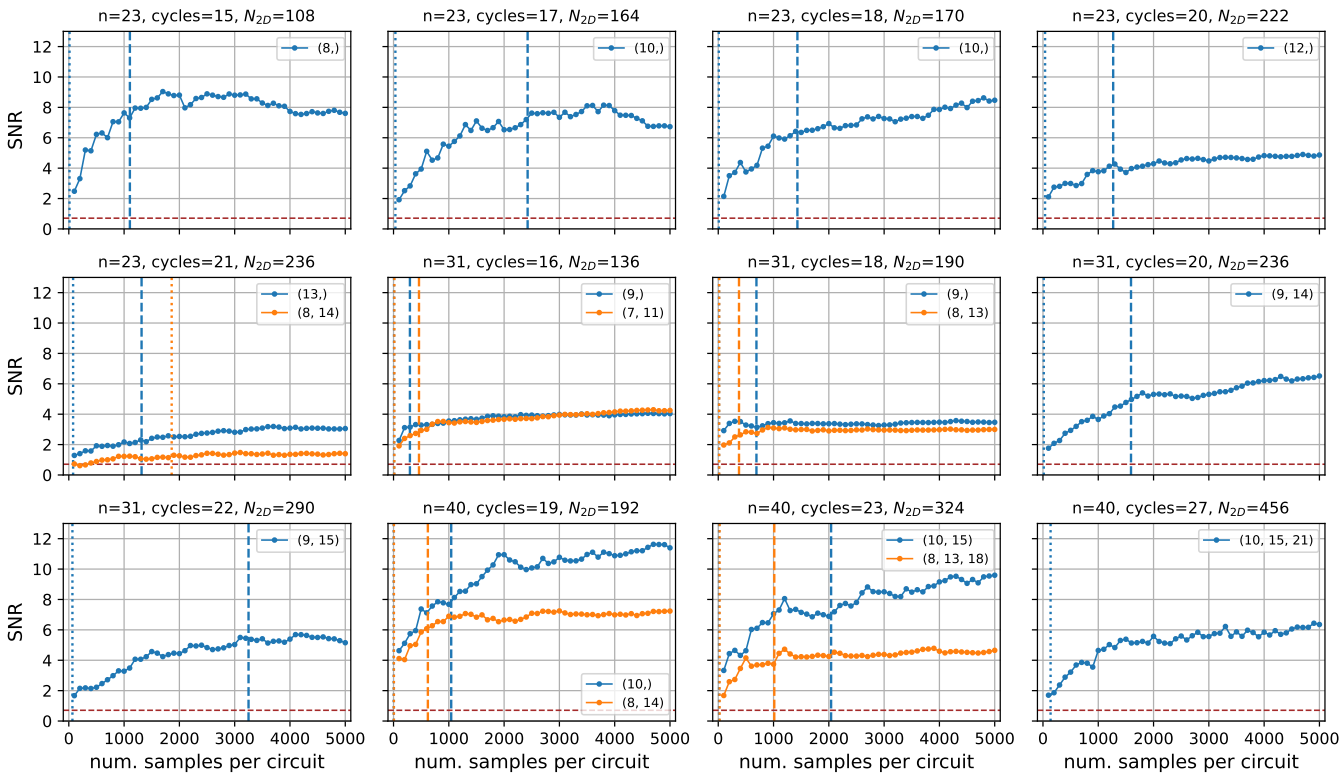


FIG. S28. **Convergence of TNMC with the number of samples.** Convergence of $\mathcal{C}^{(2)}_{\mathrm{TNMC}}$ as a function of the number of samples per circuit, K, for different circuit geometries of $n = 23$, 31, and 40 qubits and varying depths. The cut scheme used in each case is shown in the legend. In addition, the values of K_{onset}, after which the SNR is expected to grow, and K_{plateau}, after which the SNR is expected to plateau, are shown as dotted and dashed vertical lines, respectively. The baseline SNR of $1/\sqrt{2} \approx 0.71$ is shown as a dashed, brown line.

on small circuits for which it is possible to numerically compute the exact values of the OTOC$^{(1)}$, $C^{(2)}$. We consider a set of circuit geometries involving $n = 23$, 31, and 40 qubits. For each value of n, we consider circuits of different number of cycles and different number of gates. The largest example involves $N_{2\mathrm{D}} = 456$ iSWAP-like gates, i. e., almost half the number of two-qubit gates in the largest OTOC$^{(1)}$ circuits presented in this paper. In each case, we have a set of 25 circuit instances. For each circuit instance, we estimate $C^{(2)}$ with $K = 5000$ TNMC samples. Finally, for each circuit geometry, we consider either one or more values of T.

Fig. S26 shows the results of the TNMC computations over 12 circuit geometries. In all cases we see a clear correlation between the approximate TNMC OTOC$^{(1)}$ values, $\mathcal{C}^{(2)}_{\mathrm{TNMC}}$, and the exact values $\mathcal{C}^{(2)}$. The values of the SNR are typically relatively large, reaching a maximum of 11.40 for a 40 qubit circuit geometry with 19 cycles and $N_{2\mathrm{D}} = 192$, requiring only a single cut after cycle 10 in U. On the largest circuit geometry studied, with 27 cycles and $N_{2\mathrm{D}} = 456$, TNMC with three cuts (on cycles 10, 15, and 21) reaches an SNR of 6.35. Interestingly, there are non-negligible variations in the SNR achieved by TNMC for different circuit geometries and for different cut schemes. In particular, in the case of 23 qubits and 21 cycles TNMC reaches lower values of the SNR, with a value as low as 1.40 with two cuts.

In an attempt to correlate the performance of TNMC to circuit parameters, we consider the SNR of all examples presented here as a function of N_{2D}, as well as a function of the typical size of the OTOC signal, i. e., the standard deviation over circuits $\sigma(C^{(2)})$. We show this in Fig. S27. Overall there is no clear correlation between either of these quantities. When postselecting to either $n = 23$ or $n = 40$, there is a weak correlation indicating that the SNR decreases with increasing N_{2D}. Similarly, there is weak evidence that the SNR increases with signal size $\sigma(C^{(2)})$. So far, however, we have ignored the effect of having an insufficient number of Monte Carlo samples, K, for the estimation of $C^{(2)}_{\mathrm{TNMC}}$.

Convergence of TNMC with the number of samples — It is important to study the convergence of TNMC as a function of the number of Monte Carlo samples K. Fig. S28 shows the SNR as a function of the number of samples used to estimate $\mathcal{C}^{(2)}_{\text{TNMC}}$, K. The behavior of this function is as expected. At small K, the SNR is close to $1/\sqrt{2}$, shown as a dashed brown line. This is the baseline SNR, observed between two uncorrelated sets of values. Then, as K grows we observe a steady increase in SNR with some statistical fluctuations. Finally, at large K the SNR plateaus at its final value. While most examples considered show convergence (they reach their plateau value), some would need larger K to converge.

Let us now define estimate the number of samples needed for convergence of the SNR in K. Let us start by defining some quantities. First, we refer to the TNMC estimate of the OTOC$^{(1)}$ for circuit instance i as $\mathcal{C}^{(2)}_{\text{TNMC,i}}$. We also refer to the standard deviation of the distribution of Monte Carlo sample contributions to $\mathcal{C}^{(2)}_{\text{TNMC,i}}$ as $\sigma_{\text{TNMC}}(\mathcal{C}^{(2)}_{\text{TNMC},i})$. This is not to be confused with $\sigma(\mathcal{C}^{(2)}_{\text{TNMC}})$, which is the standard deviation of $\mathcal{C}^{(2)}_{\text{TNMC}}$ across circuit instances. In addition, we define the absolute value of the difference between the TNMC estimate and the true value of an OTOC as $\Delta_{\text{TNMC}}\mathcal{C}^{(2)} = |\mathcal{C}^{(2)} - \mathcal{C}^{(2)}_{\text{TNMC}}|$. Finally, follow the convention of adding a superscript s to denote standardization of the values of the OTOC to having mean 0 and standard deviation 1. With this nomenclature at hand, let us focus on three helpful quantities, all of them relative to standardized signals:

1. The signal size, defined as

$$\alpha = \sigma(\tilde{\mathcal{C}}^{(2,s)}) \tag{39}$$

   By definition of standardization, $\alpha = 1$.

2. The typical spread of TNMC Monte Carlo sample contributions, characterized by the mean value over circuit instances of $\overline{\sigma_{\text{TNMC}}(\mathcal{C}^{(2,s)}_{\text{TNMC}})}$, i.e.,

$$\beta = \overline{\sigma_{\text{TNMC}}(\mathcal{C}^{(2,s)}_{\text{TNMC}})} \tag{40}$$

3. The typical discrepancy between the TNMC estimate and the true value of the OTOC, characterized by the mean over circuit instances of $\Delta_{\text{TNMC}}\mathcal{C}^{(2,s)} = |\mathcal{C}^{(2,s)} - \mathcal{C}^{(2,s)}_{\text{TNMC}}|$, i.e.,

$$\gamma = \overline{\Delta_{\text{TNMC}}\mathcal{C}^{(2,s)}} \tag{41}$$

We are now equipped to compare the typical statistical error in the Monte Carlo estimates of the OTOC to the signal size and to the typical difference between the estimated OTOC values and the exact ones. The statistical error is typically of the order of $\beta/\sqrt{K}$, where K is the number of Monte Carlo samples. If $\beta/\sqrt{K} > \alpha = 1$, then the error in the estimates is too large to detect much correlation between $\mathcal{C}^{(2,s)}_{\text{TNMC}}$ and $\mathcal{C}^{(2,s)}$. We expect SNR to increase with K once $K \gtrapprox (\beta/\alpha)^2 = \beta^2 \equiv K_{\text{onset}}$. Similarly, we expect to approach the plateau in SNR once the statistical error in $\mathcal{C}^{(2,s)}_{\text{TNMC}}$ is smaller than the typical difference between both signals, i. e., when $\beta/\sqrt{K} \lessapprox \gamma$ or $K \gtrapprox (\beta/\gamma)^2 \equiv K_{\text{plateau}}$.

In Fig. S28 we plot vertical lines at both K_{onset} (dotted) and K_{plateau} (dashed). While both values are only ballpark indicators of the crossover between the behavior of the SNR as a function of K, we can use them as proxies to estimate the number of samples needed to measure correlation and final SNR value when comparing TNMC to exact numerics or experiments.

Performance of TNMC over large circuits — We now turn our attention to the performance of TNMC over circuits that are impractical to simulate exactly with classical computers. In those cases we do not have exact values of the OTOC to compare against, but rather experimental values.

We consider two circuit geometries involving (1) 67 qubits and $N_{2\text{D}} = 698$ gates, (2) 67 qubits and $N_{2\text{D}} = 834$, and (3) 95 qubits and $N_{2\text{D}} = 1000$ gates. In the first case we run TNMC with 4 cuts, achieving an SNR of 1.34 when comparing to experimental data. In the second case the SNR drops to 1.22 with 4 cuts, likely due to the increased circuit size. In the last case we run TNMC with 3 (on NVIDIA servers), 4, and 5 cuts, achieving SNRs of 0.74, 0.75, and 0.72, respectively. In all 3 cases the SNR is not substantially higher than baseline $1/\sqrt{2} \approx 0.71$. This data is presented in Fig. S29.

In order to compare the performance of TNMC over small and large circuits on an equal footing, we compute SNR over all circuits with respect to experimental values of the OTOC. We compile this and other data in Table SIII. It is interesting to compare the degradation of the SNR when computed with respect to experimental OTOCs instead of against exact values of the OTOCs. Beyond 40 qubits we cannot compute the SNR with respect to $\mathcal{C}^{(2)}$. We still see, however, the decrease of SNR as the circuit size increases, as well as a typically in-

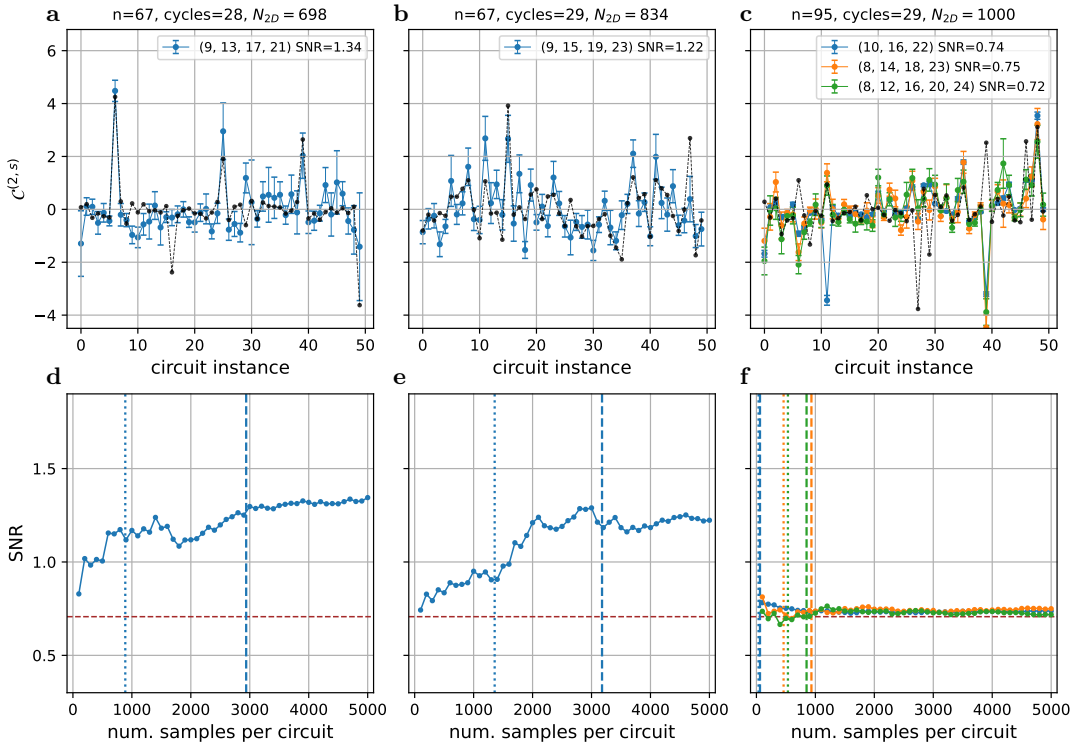


FIG. S29. **Performance of TNMC over large experiments. a-c**, Standardized TNMC approximation of $\mathcal{C}^{(2,s)}_{\mathrm{TNMC}}$ as a function of circuit instance *compared to experimental data* (black). We consider three experiments: (**a**) over 67 qubits and $N_{2D} = 698$, (**b**) over 67 qubits and N_{2D}, and (**c**) over 95 qubits and $N_{2D} = 1000$. The cut schemes and SNR *with respect to experimental data* are shown in the legend. Note the presence of a significant correlation in the first case, with an SNR of 1.55, and the lack of a meaningful correlation in the second case. In both cases we use 5000 Monte Carlo samples per point. **d-f**, Convergence of TNMC with number of samples up to $K = 5000$. The degradation of SNR compared to the values of Fig. S26 is due to the fact that here we compare against experimental data, which is itself imperfect, and to the increased size of the circuits in terms of qubits, depth, and N_{2D}. The SNR decreases as the circuits grow in volume. In addition, at $n = 95$ and $N_{2D} = 1000$ we see a slight increase in the SNR as the number of cuts decreases, although remaining small and close to $1/\sqrt{2}$.

creasing SNR as the number of cuts decreases. Finally, at $n = 95$ and $N_{2D} = 1000$ we cannot achieve a substantially large SNR.

One possible strategy to find statistically significant SNR on the $N_{2D} = 1000$ circuits could be attempting to run TNMC with 2 cuts. After optimization of the TNCs involved in this procedure, we estimate that a single sample with $T = 2$ (optimally chosen at depths 12 and 20) would require 1.81×10^{15} FLOP per sample. This is equivalent to effectively 5.3 seconds for every 1000 samples on Frontier, which delivers a peak performance of 1.71 ExaFLOPS. We constrain the memory footprint of each TNC to satisfy that of a single GPU. Different independent contractions, such as those of different samples, can be carried out in parallel over different GPUs. This is a somewhat challenging computational effort.

Future TNMC improvements — It is possible to improve the TNMC method and its implementation to achieve lower computational cost. First, the current implementation cuts the OTOC unitary over all qubits at certain depth locations. More flexible cutting schemes can certainly be beneficial. Furthermore, the sampling procedure of Ref. [38] allows for more elaborate reorganization of the tensor network evaluations involved, leading to a lower cost per sample than the current implementation. In addition, the evaluation of Eq. (29) and similar might admit approximations that lower the computational cost. Finally, considering contractions with less stringent memory constraints can lower their cost dramatically. This, however, requires a substantial effort in the underlying TNC implementation. These potential improvements are left as directions of future research.

n	Cycles	N_{2D}	Cuts	TNMC (Exact)			TNMC (Experiment)			Experiment (Exact)
				K_{onset}	K_{plateau}	SNR	K_{onset}	K_{plateau}	SNR	SNR
23	15	108	(8)	13	1105	7.61	13	479	4.74	7.21
	17	164	(10)	34	2425	6.73	34	575	3.08	5.26
	18	170	(10)	10	1432	8.47	10	495	5.34	6.67
	20	222	(12)	38	1270	4.86	48	493	2.55	3.57
	21	236	(13)	78	1316	3.06	78	709	2.46	2.45
			(8, 14)	1860	7981	1.40	1861	4344	1.13	
31	16	136	(9)	9	296	4.04	9	127	2.87	4.23
			(7, 11)	15	460	4.25	15	176	2.79	
	18	136	(9)	14	691	3.47	14	313	3.45	3.85
			(8, 13)	17	375	3.00	17	197	2.40	
	20	236	(9, 14)	15	1592	6.52	15	466	3.46	3.72
	22	290	(9, 15)	65	3250	5.15	65	455	2.19	3.40
40	19	192	(10)	4	1043	11.40	4	188	5.41	4.88
			(8, 14)	7	621	7.24	7	317	5.53	
	23	324	(10, 15)	15	2041	9.60	15	504	4.53	3.88
			(8, 13, 18)	24	1013	4.65	24	291	2.75	
	27	456	(10, 15, 21)	138	7863	6.35	138	1623	2.78	2.59
67	28	698	(9, 13, 17, 21)	-	-	-	684	2347	1.34	-
67	29	834	(9, 15, 19, 23)	-	-	-	1357	3178	1.22	-
95	29	1000	(10, 16, 22)	-	-	-	38	70	0.74	-
			(8, 14, 18, 23)	-	-	-	279	602	0.75	-
			(8, 12, 16, 20, 24)	-	-	-	506	775	0.72	-

TABLE SIII. **Comparison between TNMC, experimental, and exact signals.** Analysis of the performance of TNMC compared to experimental and exact OTOC signals. We show the number of qubits n, the number of cycles in U, the number of iSWAP-like gates, and the cutting scheme used by TNMC. The results include the SNR between the TNMC OTOC values and the exact ones (when available), as well as the SNR between TNMC and experimental values, and that between the experiment and the exact values. In addition, we estimate K_{onset} at which some non-trivial SNR is expected and K_{plateau} at the final plateau in SNR is approached for TNMC. We consider the same circuit geometries as in Figs. S26 and S29. The last two rows correspond to large circuits for which only experimental values of the OTOC are available.

4. *Clifford-based expansions (CBE)*

Clifford-base expansion (CBE) algorithms has been successfully used in the past to classically simulate $\text{OTOC}^{(1)}$ circuits [1]. Similar to the cached Monte Carlo, the idea of (CBE) algorithms is to follow the evolution of the coefficients b_P, and keep track of only those coefficients which are non-zero. Unlike cached Monte Carlo, coefficient are explored "depth-first": that is, a single branch of the Pauli-path is followed till all u_t are applied, and only the final Pauli string is stored in memory. Without any truncation, CBE algorithms provide the exact calculation of the correlation operator $\mathcal{C}^{(2k)}$. However, the number of b_P coefficients exponentially grows with the number of non-Clifford gates.

In [1], the only non-Clifford gates used for $\text{OTOC}^{(1)}$ were 4 out of 8 single-qubit rotations, i.e. $\sqrt{W^{\pm 1}}$ and $\sqrt{V^{\pm 1}}$, with $W = \sqrt{(X+Y)/\sqrt{2}}$ and $V = \sqrt{(X-Y)/\sqrt{2}}$. Because

$$
\begin{aligned}
W^\dagger X W &= \frac{X+Y}{2} + \frac{Z}{\sqrt{2}} \\
V^\dagger X V &= \frac{X-Y}{2} + \frac{Z}{\sqrt{2}} \\
W^\dagger Y W &= \frac{X+Y}{2} - \frac{Z}{\sqrt{2}} \\
V^\dagger Y V &= \frac{X-Y}{2} - \frac{Z}{\sqrt{2}} \\
W^\dagger Z W &= \frac{Y-X}{\sqrt{2}} \\
V^\dagger Z V &= -\frac{X+Y}{2},
\end{aligned}
$$

the expected number of paths to explore will be of the order of $\sim 3^{N_D/2} 2^{N_D/4}$, with N_D being the number of non-Clifford rotations (see Fig. S15 of [1]). In our OTOC

experiments, all gates are non-Clifford, making the exact simulation of $\mathcal{C}^{(2k)}$ unpractical.

In parallel to the cached Monte Carlo, approximation schemes can be introduced to CBE algorithms to reduce the amount of coefficients b_P to keep track of. One possible approximation consists in only removing all paths with coefficients b_P below a certain threshold. By properly tuning such a threshold, it is possible to reduce the amount of paths to explore. However, our numerical studies (not reported) show that the cached Monte Carlo largely outperforms CBE.

C. Classical simulation of higher-order OTOCs

In this section we expand our OTOC simulation efforts to target OTOC$^{(2)}$, both attempting to extend the methods attempted for OTOC$^{(1)}$, and trying new methods solely targeting $\mathcal{C}^{(4)}_{\text{off}-\text{diag}}$. Despite trying many approaches, we have been unable to find a polynomially-scaling method with significant SNR (> 1) for OTOC$^{(2)}$, which marks a clear difference to the success of Monte Carlo methods on $\mathcal{C}^{(2)}$. This underpins our claim that the experimental data reported in the main text is beyond-classical. We begin in Sec. III C 1 by attempting to extend the vanilla Monte Carlo to estimate OTOC$^{(2)}$. Beyond small system sizes, we find that SNR above 1 can not be achieved. The same is true for the cached Monte Carlo, which we attempt in Sec. III C 2 and find evidence of a clear sign problem. Indeed, this sign problem persists even when estimating the mean OTOC$^{(2)}$, $\overline{C^{(4)}}$, as we show in Sec. III C 3, and to a much stronger degree than the sign problem in estimating $\sigma(C^{(2)})$ that was reported on in Ref. [1].

We next turn to exponentially-scaling methods, or methods with a formal exponential cost that may be heuristically truncated (e. g. by bond dimension). This includes many popular methods such as Clifford expansions III B 4, Pauli-weight truncation algorithms III C 6, and variational approaches such as neural quantum states III C 8 and matrix-product states III C 7. None of the above are able to achieve a significant SNR in OTOC$^{(2)}$ approximation with the resources at hand.

Therefore, we believe that tensor network contraction remains the fastest classical algorithm for OTOC$^{(2)}$ simulation. We detail methods for the optimization of exact tensor network contraction for OTOC$^{(2)}$ circuits in Sec. III C 4.

Then, in an attempt to reduce the contraction costs, in Sec. III C 5 we detail a heuristic method to remove the least important gates at the cost of reducing the simulation SNR. It is challenging to estimate the resulting tensor contraction cost after such gate removal protocol, since it is hard to extrapolate the SNR as a function of the removed gate count in beyond-classical circuits; our best estimates yield a range of between a few minutes and a month. However, as the true number of removable gates remains unknown, we keep our claim that our OTOC$^{(2)}$ experiment is beyond-classical.

1. *Vanilla Monte Carlo simulation of OTOC$^{(2)}$*

In this section we adapt the methods of the previous section that were used for simulating the OTOC$^{(1)}$ to simulating the OTOC$^{(2)}$. For simplicity, here we consider splitting the circuit U into only two parts $U = u_1 u_2$. Writing an observable B as $B(t=1) = u_1^\dagger B u_1$, we obtain as an expression for the OTOC$^{(2)}$

$$\mathcal{C}^{(4)} = \left\langle [u_2^\dagger B(1) u_2 M]^4 \right\rangle. \tag{42}$$

Now, let us expand the time evolved butterfly in terms of Pauli strings P, yielding

$$\mathcal{C}^{(4)} = \sum_{P,P',Q,Q'} b_P(1) b_{P'}(1) b_Q(1) b_{Q'}(1) \left\langle u_2^\dagger P u_2 M u_2^\dagger P' u_2 M u_2^\dagger Q u_2 M u_2^\dagger Q' u_2 \right\rangle. \tag{43}$$

As in Section III B 1, we can square the $b_P(1)$ terms to give a probability distribution $p_P(1) = |b_P(1)|^2$, which corresponds to the weight of the string P in the decomposition of B.

To obtain a positive Monte Carlo measure, we need to disregard the terms in Eq. (43) which do not allow for a probability interpretation in this basis. This yields the approximation

$$\mathcal{C}^{(4)} \approx \mathcal{C}^{(4)}_{\text{MC}} = \sum_{P,P',Q,Q'} (\delta_{P,P'}\delta_{Q,Q'} + \delta_{P,Q}\delta_{P',Q'} + \delta_{P,Q'}\delta_{P',Q} - 2\delta_{P,P'}\delta_{P',Q}\delta_{Q,Q'})\, b_P(1) b_{P'}(1) b_Q(1) b_{Q'}(1) \times \left\langle u_2^\dagger P u_2 M u_2^\dagger P' u_2 M u_2^\dagger Q u_2 M u_2^\dagger Q' u_2 \right\rangle, \tag{44}$$

where we subtract the term where all Pauli strings are the same, as this is included across the first three terms. The terms in the bracket give four contributions to the MC-approximated OTOC, which we label in order as "AABB", "ABAB", "ABBA", and "AAAA". We can evaluate each contribution separately by Monte Carlo

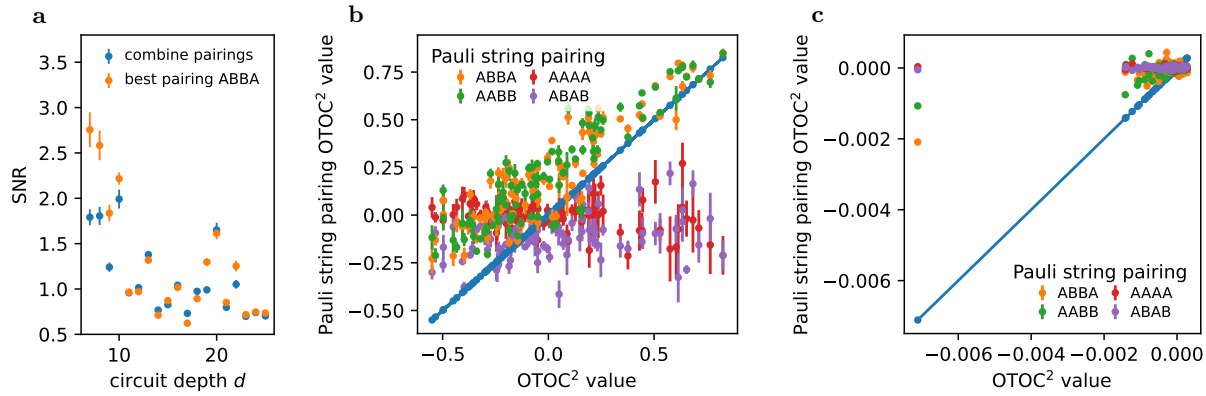


FIG. S30. **a**, The results for the SNR when approximating the signal using the approximation introduced in Eq. (III C 1). We also consider the ABBA pairing. This pairing which yields the highest SNR in 100 random circuit instances. **b-c**, the true OTOC$^{(2)}$ expectation values for 100 random circuit instances plotted against the different pairings Eq. (III C 1). The circuit depth is 10 (center) and 20 (right), the system size is 26.

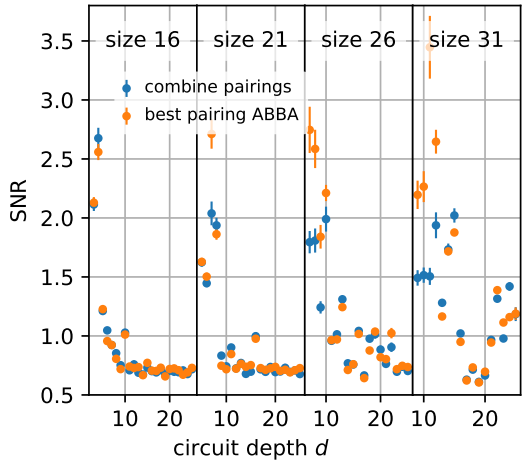


FIG. S31. **SNRs when using the diagonal parts of the sum to estimate the OTOC$^{(2)}$ signal, but only considering the results inside a 4σ region around the mean to exclude outliers.** See also Fig. S30. The number of qubits, the sizes, are 16, 21, 26 and 31 from left to right.

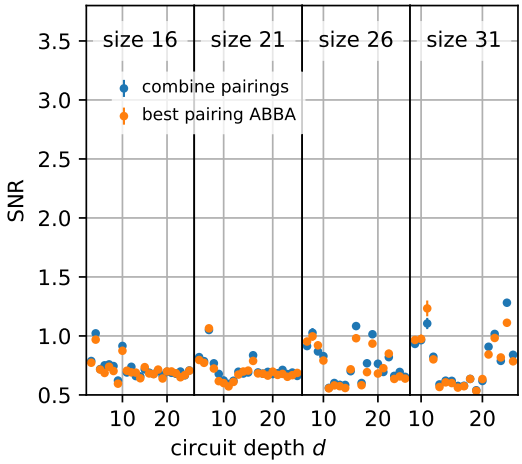


FIG. S32. **SNR when using the diagonal pairings to estimate the subtracted OTOC$^{(2)}$ signal, i. e. the true signal with the ABBA pairing subtracted.** We only consider the results inside a 4σ region around the mean. The number of qubits are 16, 21, 26 and 31 from left to right.

sampling from the $\boldsymbol{p}(1)$ vectors and evaluating the expectation value in Eq. (III C 1) as an OTOC$^{(2)}$ (possibly with different butterfly operators at different times).

We test this approximation numerically by considering a rectangular $l \times 5$ system, with the four corner qubits removed, for $l = 4$, 5, 6 and 7. In Fig. S31, we plot the SNR of the approximation defined in Eq. (III C 1) (using only a single cut), as a function of the depth. For small depths, we observe that this approximation can be used to approximate the OTOC$^{(2)}$. At larger depths, there may be individual circuits with large signal sizes, yielding a large SNR despite the approximation being poor for the most circuits, see Fig. S30.

In Fig. S31, we plot the contribution to $\mathcal{C}^{(4)}_{\mathrm{MC}}$ from various pieces of Eq. (III C 1). We observe that keeping only the ABBA pairing works comparably to keeping a larger part of the sum. Furthermore, we note that the two pairings ABBA and AABB have the largest contributions to the OTOC$^{(2)}$ signal, see Fig. S30.

Due to the strong correlation between the whole sum OTOC$^{(2)}$ and the pairing ABBA at small depths, we look into the sum in Eq. (43) without the pairing term ABBA. This can be done on the quantum device by measuring the OTOC$^{(2)}$ and the ABBA term and taking the dif-

ference; we now study this option numerically. Given the chance of outliers, we only consider circuits generating the subtracted $\mathrm{OTOC}^{(2)}$ signals inside a 4-σ range around the mean of the $\mathrm{OTOC}^{(2)}$, see Fig. S30. The results for the different system sizes are shown in Fig. S32. We observe that the SNR drops, and the Monte Carlo is unable to estimate the residual correlation. Though subtracting the ABBA part induces some correlation, we cannot observe any systematic behavior. We conclude that these two parts of the sum are fundamentally different. One part can be estimated, by restricting Pauli strings to be the same after half the iteration, while the other part is far less correlated with those terms.

Note that simulating the ABBA pairing is not necessarily computationally simple, and should scale exponentially with system size and circuit depth. However, it is exponentially cheaper than simulating the whole circuit, and may be feasible with sufficient computational power.

2. *Cached Monte Carlo and the sign problem*

We now attempt to go beyond the classical pairings shown in Section III C 1 and see if cached Monte Carlo may be used to compute the off-diagonal contributions to $\mathrm{OTOC}^{(2)}$. To this end, let us discuss a natural extension of the projection procedure Eq. (28) onto the $\mathrm{OTOC}^{(2)}$ computation:

$$\mathcal{C}^{(4)} = \langle 0|(B(T)M)^4|0\rangle, \tag{45}$$

which now contains four copies of $B(T)$. At first, one sparsely time-evolves a single copy of the butterfly operator until a cache overfills at time t. In a natural generalization of the Eq. (28) we decompose each of the four caches (before the first projection, they are identical) as

$$B_k(t) = \sum_{\alpha_k} z_{k,\alpha_k} P_i^{\alpha_k} \otimes B_{k,\bar{i},\alpha_k}, \tag{46}$$

where, $P_i^{\alpha_k}$ is a Pauli operator acting on a site i, $B_{k,\bar{i},\alpha_k}$ is the "adjoint" operator defined at all sites except for i, $0 \leqslant k < 4$ enumerates the four caches, and the prefactors are normalized as $\sum_{\alpha_k} z_{k,\alpha_k}^2 = 1$.[6] Plugging this decomposition in, we obtain

$$\mathcal{C}^{(4)} = \sum_{\alpha_1,\alpha_2,\alpha_3,\alpha_4} \underbrace{\left(\prod_{k=0}^{3} z_{k,\alpha_k}\right)}_{W(\boldsymbol{\alpha})} \times \tag{47}$$
$$\times \langle 0| \prod_{k=0}^{3} \left(U^\dagger(t,T) P_i^{\alpha_k} \otimes B_{\bar{i},\alpha_k} U(t,T) M\right) |0\rangle,$$

Unlike the $\mathrm{OTOC}^{(1)}$ case, where the trace condition only allows equal Pauli operators, here, of those $4^4 = 256$ terms, 64 satisfy

$$\mathrm{Tr} \prod_k P_i^{\alpha_k} \neq 0, \tag{48}$$

including contributions with $\alpha_1 \neq \alpha_2 \neq \alpha_3 \neq \alpha_4$ (that is, all four different). Having four copies of the butterfly operator leads to the *sign problem* manifested in the following way. On average, $z_{k,\alpha_k} \approx 1/2$, and a typical weight $W(\boldsymbol{\alpha}) \sim 1/16$. But since there are 64 non-zero contributions, $W(\boldsymbol{\alpha})$ does not form a probability distribution.

Naturally, one could neglect more terms, by requesting pairwise matching $P_0 = P_1$, $P_2 = P_3$ (or any permutation thereof). Being restricted to these 16 contributions, $W_{\mathrm{restricted}}(\boldsymbol{\alpha})$ forms a probability distribution. This restriction represents the computation of the *diagonal* part of $\mathrm{OTOC}^{(2)}$.

Without such a restriction, one needs to re-weight the Monte Carlo measure by $\sum_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) \approx 4$. Such reweighting rescales an observable by a factor ~ 4 each time a projection is done, and leads to the exponential variance growth. This remanifestation of the sign problem for $\mathrm{OTOC}^{(2)}$ prevents the cached Monte Carlo from accessing the non-diagonal part of $\mathrm{OTOC}^{(2)}$.

3. *Sign problem for $\overline{\mathcal{C}^{(4)}}$-preserving dynamics*

Unlike the case of $\mathcal{C}^{(2)}$, we find that Monte Carlo sampling is stymied by a sign problem even for the simpler problem of computing the average value $\overline{\mathcal{C}^{(4)}}$ over a Haar-random ensemble of gates. In this section we show that the computation of $\overline{\mathcal{C}^{(4)}}$ can be viewed as a globally truncated form of the dynamics similar to that achieved by vanilla Monte Carlo for $\mathcal{C}^{(2)}$, which, if not for the sign problem, would provide a starting point for approximating individual circuit $\mathcal{C}^{(4)}$ values. Thus, it is important to verify that this sign problem is actually a significant barrier. To quantitatively measure the severity of this

[6] In practice, such decomposition is done by selecting all Pauli strings from the cache $\mathcal{C}$ that have a Pauli P_i at a site i.

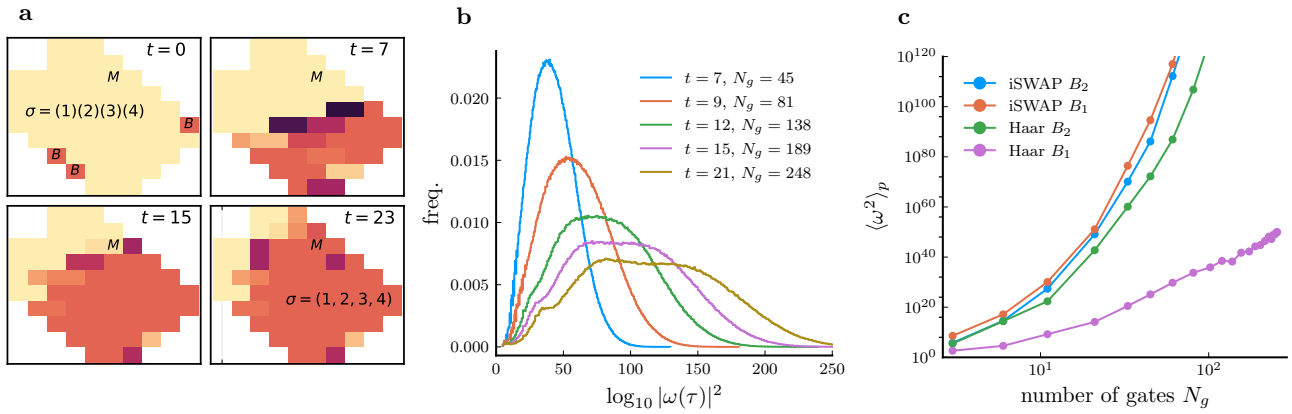


FIG. S33. **Sign problem for computing $\overline{\mathcal{C}^{(4)}}$.** All parts of this figure use the 65-qubit experimental $\mathcal{C}^{(4)}$ circuit layouts and either Haar-random two qubit gates or iSWAP-like gates, as indicated. **a**, A snapshot of a typical permutation basis trajectory sampled using Eq. 53. Different colors correspond to different permutations – a typical trajectory has a single large domain behind the operator front, which progresses outwards with the butterfly velocity. **b**, The distribution of trajectory weights with iSWAP-like gates, cut at different depths t after which the circuit has experienced N_g total gates. In a large system, the distribution approaches log-normal with a mean and width that grows in time (not shown); the behavior here is similar but with distortions due to the boundaries of the circuit. **c**, Variance of trajectory weights for a sample of $3 \cdot 10^6$ trajectories, using both iSWAP-like gates and Haar random two qubit gates, with two different choices for the basis in which the sampling takes place [see text]. In all cases, the weights grow tremendously quickly, indicating that sampling has a severe sign problem.

sign problem, we consider sampling techniques that sample from a related positive distribution and absorb the sign into sample weights, as is done by quantum Monte Carlo practitioners. With this approach, we find that the growth of classical complexity is exponential in the volume of the operator front of $B(t)$, rather than the total volume of the operator $B(t)$, a reduction that is enabled by discarding irrelevant components of 4 copies of the butterfly operator $B^{\otimes 4}(t)$. However, this growth rate is still more than sufficient to render the circuits considered in this paper beyond any classical compute resources.

In this section we use a 'folded' picture of the dynamics, in which 4 copies of the butterfly operator B are simultaneously evolved in an expanded Hilbert space of $4^4 = 256$ operators per site. Using a trace over 4 copies of the original Hilbert space, we can rewrite the definition of $\mathcal{C}^{(4)}$ as

$$\begin{aligned}\mathcal{C}^{(4)} &= 2^{-N}\mathrm{Tr}\left[\Pi^{\dagger}_{(1,2,3,4)} B^{\otimes 4}(t) M^{\otimes 4}\right] \\ &= 2^{-N}\mathrm{Tr}\left[\Pi^{\dagger}_{(1,2,3,4)} U(t)^{\otimes 4} B^{\otimes 4} U^{\dagger\otimes 4}(t) M^{\otimes 4}\right]\end{aligned}$$

Here, $\Pi_{(1,2,3,4)}$ refers to a global permutation operator that permutes the 4 copies of the Hilbert space, and we use the standard cycle notation for permutations — thus, $(1,2,3,4)$ refers to a permutation that sends copy j to copy $(j+1)(\mathrm{mod}4)$. We note that the $4! = 24$ global permutations commute with the dynamical operator $U(t)^{\otimes 4}$ as it is the same on all 4 copies.

Naively, this expanded Hilbert space greatly increases the complexity of dynamics algorithms, but it also enables truncating much of the information encoded in the combined 4-copy operator $B^{\otimes 4}(t)$ while still approximately capturing the $\mathcal{C}^{(4)}$ front. To help explain the origins of this simplification, we note that this expression can be viewed as analogous to a time-ordered, equilibrium correlation function in the enlarged Hilbert space. Just as in that case, this correlation function undergoes a process of thermalization, in which the information content of $B^{\otimes 4}(t)$ becomes scrambled and inaccessible to local observables. For sufficiently deep circuits, $B^{\otimes 4}(t)$ locally approaches equilibrium. Thermalization of a Heisenberg-evolved operator effectively projects the operator onto the global symmetries of the evolution as equilibrium is approached; components of $B^{\otimes 4}(t)$ on these global symmetry operators are fixed in time, while all other components effectively decay away. The global symmetries of the 4 copy evolution are in fact just the 24 global permutation operators. The process of thermalization occurs behind the $\mathcal{C}^{(4)}$ operator front, as the equilibrium value of $\mathcal{C}^{(4)}$ is 0. The benefit of exploiting thermalization is thus to discard information in $B^{\otimes 4}(t)$ behind the operator front.

In this section, we specifically want to compute the average value $\overline{\mathcal{C}^{(4)}}$ over a Haar random ensemble of single qubit gates inserted at every qubit and at every time step.

This average is equivalent to projecting the dynamics of $B^{\otimes 4}(t)$ locally into the span of the permutation operators on the 4-copies of each site. These projections are sufficient to truncate globally scrambled information in $B^{\otimes 4}(t)$ — the average dynamics is non-unitary and explicitly drives the operator into a superposition of the 24 global symmetries behind the front, with only information in the front being non-trivial.

By construction, computing this average cannot reproduce the spread in $\mathcal{C}^{(4)}$ values for circuits that have the same two-qubit gates and only differ by single qubit gates. However, one can easily imagine improvements that can capture more variance between circuits by restricting the projections to a subset of space-time locations, at increased computational complexity. The cached Monte Carlo and TNMC algorithms for computing $\mathcal{C}^{(2)}$ can also be viewed as computing the result of projections at subsets of space-time locations, with those projections also resulting from averaging over an ensemble of single qubit gate insertions; just as those algorithms improve upon vanilla Monte Carlo, achieving some success in approximating $\mathcal{C}^{(2)}$, we expect that similar improvements can be made upon any algorithm that computes $\overline{\mathcal{C}^{(4)}}$. As these improvements use increased computational cost to capture more of the variance between circuits, we expect that the complexity of computing this average $\overline{\mathcal{C}^{(4)}}$ provides a lower bound for the difficulty of this entire class of algorithms.

Now let us explain the mechanics of computing

$$\overline{\mathcal{C}^{(4)}} = 2^{-N}\mathrm{Tr}\left[\Pi^{\dagger}_{(1,2,3,4)}\overline{B^{\otimes 4}(t)}M^{\otimes 4}\right]$$

by evolving the average 4-copy operator $\overline{B^{\otimes 4}(t)}$. This calculation follows the strategy used for several random circuit calculations in the literature. As the average over each single qubit gate insertion is independent, we can replace each two-qubit gate u in the circuit, which appears in this operator evolution as a superoperator $S[u] = u^{\otimes 4}\cdot u^{\dagger\otimes 4}$, with a rotation averaged superoperator

$$T = \int dv_1 dv_2 dv_3 dv_4 S\left[(v_1\otimes v_2)u(v_3\otimes v_4)\right].$$

As stated before, the operation of T is precisely only nonzero on the span of the Haar-invariant operators, which are spanned by products of permutation operators on each site, and is zero on the orthogonal complement to this space.

To proceed further, we pick a basis of operators for the span of the permutation operators, and expand our dynamics into a sum over basis trajectories. In our numerical experiments, the best choice of basis is the local permutation operators $\{\pi_\sigma\}$ which permute the 4 copies of one site as the basis elements, which has the unique property that the fixed points of the average operator evolution can each be represented by a single computational basis element, rather than a large superposition. This basis is non-orthogonal and overcomplete, as only 14 of the 24 permutations are linearly independent. I will assume this choice below to simplify the exposition, but alternate choices proceed similarly.

Let us consider directly expanding the dynamics as a sum over permutation basis trajectories

$$\begin{aligned}
\mathcal{C}^{(4)} &= 2^{-N}\sum_{\boldsymbol{\tau}} T_{\boldsymbol{\tau}} B_{\boldsymbol{\tau}^0} M_{\boldsymbol{\tau}^f}, \qquad (49)\\
T_{\boldsymbol{\tau}} &\equiv \prod_{t=0}^{t_f-1}\prod_{\langle i,j\rangle\in G(t)} \hat{T}^{\sigma_i^{t+1}\sigma_j^{t+1}}_{\sigma_i^t,\sigma_j^t},\\
B(0)^{\otimes 4} &= \sum_{\boldsymbol{\tau}^0=\{\sigma_1^0,\ldots\sigma_N^0\}} B_{\boldsymbol{\tau}^0}\pi_{\sigma_1^0}\ldots\pi_{\sigma_N^0}\\
M_{\boldsymbol{\tau}^f} &= \mathrm{Tr}\left[\Pi^{\dagger}_{(1,2,3,4)}M^{\otimes 4}\pi_{\sigma_1^{t_f}}\ldots\pi_{\sigma_N^{t_f}}\right]
\end{aligned}$$

where the trajectory label $\boldsymbol{\tau} = \{\sigma_1^0,\sigma_2^0,\ldots\sigma_N^{t_f}\}$ labels each assignment of permutations to all sites and times, $T_{\boldsymbol{\tau}}$ labels the weight assigned to this trajectory, and $B_{\boldsymbol{\tau}^0}, M_{\boldsymbol{\tau}^{t_f}}$ quantify the effect of the initial and final boundary conditions, respectively.

The final boundary condition produces trajectory weights that can be as large as 2^{3N} from the overlap between a permutation $\pi_{(1,2,3,4)}$ on each site and $\Pi^{\dagger}_{(1,2,3,4)}$. As $\mathcal{C}^{(4)}$ is a quantity bounded between -1 and 1, this implies that vary rare trajectories could make order 1 contributions, leading to difficulties in sampling. As such, we find that the following rescaling of the basis elements fixes the problem

$$\pi_\sigma \to \frac{\pi_\sigma}{\mathrm{Tr}(\pi^{\dagger}_{(1,2,3,4)}\pi_\sigma)},$$

rendering each $M_{\boldsymbol{\tau}^f}$ to be of order one. Moreover, the same replacement allows the 2^{-N} factor in Eq. 49 to be absorbed into the butterfly operator coefficients $B_{\boldsymbol{\tau}^0}$, yielding order one numbers for those as well. Finally, because of the rescaling of the permutations, we have conservation of total output weight at each gate:

$$\sum_{\sigma_i^{t+1},\sigma_j^{t+1}} \hat{T}^{\sigma_i^{t+1}\sigma_j^{t+1}}_{\sigma_i^t,\sigma_j^t} = 1. \qquad (50)$$

This means that the total weight is conserved throughout the evolution, encoding the conservation law

$$2^{-N}\mathrm{Tr}\left[\Pi^{\dagger}_{(1,2,3,4)}B^{\otimes 4}(t)\right] = \frac{\mathrm{Tr}(B(t)^4)}{\mathrm{Tr}(I)} = \frac{\mathrm{Tr}(B(0)^4)}{\mathrm{Tr}(I)}.$$

When $B(0)$ is a Pauli operator, this conserved value takes the value 1.

In summary, we have *almost* written the evolution of $\overline{B^{\otimes 4}(t)}$ as a Markovian evolution. However, the transfer matrix has signs, and the trajectory weights are both positive and negative, despite being conserved. We will use a reweighting scheme to try to sample anyway. To see how this works, it is useful to rewrite the observable as

$$\overline{\mathcal{C}^{(4)}} = \frac{\sum_{\boldsymbol{\tau}} T(\boldsymbol{\tau})M(\boldsymbol{\tau})}{\sum_{\boldsymbol{\tau}} T(\boldsymbol{\tau})},$$

where $T(\boldsymbol{\tau}) = T_{\boldsymbol{\tau}}B_{\boldsymbol{\tau}^0}$ and $M(\boldsymbol{\tau}) = M_{\boldsymbol{\tau}^f}$, and where the denominator

$$\sum_{\boldsymbol{\tau}} T(\boldsymbol{\tau}) = 1, \tag{51}$$

which can be derived from Eq. 50. The sampling schemes under consideration sample trajectories according to a probability distribution $p(\boldsymbol{\tau})$ and reweight the trajectories by $\omega(\boldsymbol{\tau}) = T(\boldsymbol{\tau})/p(\boldsymbol{\tau})$, so that

$$\overline{\mathcal{C}^{(4)}} = \frac{\sum_{\boldsymbol{\tau}} p(\boldsymbol{\tau})\omega(\boldsymbol{\tau})M(\boldsymbol{\tau})}{\sum_{\boldsymbol{\tau}} p(\boldsymbol{\tau})\omega(\boldsymbol{\tau})} \approx \frac{\sum_{\text{sampled } \boldsymbol{\tau}} \omega(\boldsymbol{\tau})M(\boldsymbol{\tau})}{\sum_{\text{sampled } \boldsymbol{\tau}} \omega(\boldsymbol{\tau})}.$$

The severity of the sign problem can be diagnosed by computing the sample variance of the denominator

$$\mathrm{Var}(\omega) = \sum_{\boldsymbol{\tau}} p(\boldsymbol{\tau})\omega(\boldsymbol{\tau})^2 - \left(\sum_{\boldsymbol{\tau}} p(\boldsymbol{\tau})\omega(\boldsymbol{\tau})\right)^2 = \langle \omega(\boldsymbol{\tau})^2\rangle_p - 1,$$

where $\langle\rangle_p$ indicates the average is taken with respect to the distribution p. As $\langle\omega\rangle_p = 1$ by construction, a number of samples $N_s \sim \mathrm{Var}(\omega)$ is necessary to get an accurate estimate of the denominator. As an example, in the case where

$$p(\boldsymbol{\tau}) = \frac{|T(\boldsymbol{\tau})|}{\sum_{\boldsymbol{\tau}} |T(\boldsymbol{\tau})|}, \tag{52}$$

one can show with Eq. 51 that

$$\langle\omega^2\rangle_p = \frac{1}{\langle \mathrm{sgn}\ T(\tau)\rangle_p^2};$$

so an exponentially large $\langle\omega^2\rangle_p$ exactly corresponds to an exponentially small average sign, which is commonly used as an indicator of the sign problem — see, for example Ref. [39].

Instead of sampling from Eq. 52, we use the following incremental sampling algorithm to sample trajectories $\boldsymbol{\tau}$ and corresponding weights $\omega(\boldsymbol{\tau})$ one gate at a time:

- Start with an initial configuration $(\sigma_1^0, \sigma_2^0, \sigma_3^0, \ldots)$ and weight $\omega = 1$
- Stochastically apply each gate one at a time, replacing σ_i, σ_j by σ_i', σ_j' with probability

$$p_{\sigma_i\sigma_j}^{\sigma_i'\sigma_j'} = \frac{|\hat{T}_{\sigma_i\sigma_j}^{\sigma_i'\sigma_j'}|}{\sum_{\sigma_i'\sigma_j'} |\hat{T}_{\sigma_i\sigma_j}^{\sigma_i'\sigma_j'}|}. \tag{53}$$

- Update the weight variable

$$\omega \leftarrow \omega\frac{\hat{T}}{p} = \omega\,\mathrm{sgn}\left(\hat{T}_{\sigma_i\sigma_j}^{\sigma_i'\sigma_j'}\right)\sum_{\sigma_i'\sigma_j'} |\hat{T}_{\sigma_i\sigma_j}^{\sigma_i'\sigma_j'}|.$$

At each step, the weight of the trajectory potentially changes sign and is multiplied by a factor $Z(\sigma_i, \sigma_j) = |\sum_{\sigma_i'\sigma_j'} \hat{T}_{\sigma_i\sigma_j}^{\sigma_i'\sigma_j'}| \geqslant 1$, where σ_i and σ_j are two permutations at the current location of a gate. One may expect then that the trajectory weights grow exponentially in magnitude with the number of applied gates. In practice, the growth is much slower than this implies for the following reason: $Z(\sigma_i, \sigma_j) = 1$ whenever $\sigma_i = \sigma_j$, which follows from the conservation law Eq. 50, and additionally $Z(\sigma_i, \sigma_j) = 1$ whenever σ_i and σ_j differ by a single transposition. Furthermore, constant domains of a single permutation σ are fixed under this update rule, and typical trajectories produce a large growing domain of a single permutation that expands outwards with the butterfly velocity, as illustrated in Fig. S33 (a). Thus, only gates in the region of the operator front contribute to the growth of weights $\langle\omega(\tau)^2\rangle_p$ and the severity of the sign problem. This suggests that our algorithm has successfully discarded irrelevant information in $B^{\otimes 4}(t)$ behind the front.

Numerical results for the growth of the sign problem for the 65 qubit $\mathcal{C}^{(4)}$ circuits in this paper are shown in Fig. S33 (b-c). Due to the repeated random multiplications that govern the growth of the weights, the weights follow a log-normal like distribution whose mean grows with depth of the circuit. The magnitude of this growth is rapid for $q = 2$, producing astronomically sized weights. In Fig. S33 (c), we compare the growth of the number of samples needed $N \sim \langle\omega^2\rangle_p$ versus depth for the circuits used in this paper, which use iSWAP-like gates, as well

as circuits with the same layout and Haar-random two qubit gates. The circuits of this paper produce complexity growth that is far too rapid to be overcome with any existing classical compute resources — indeed, the complexity of this algorithm far exceeds that of brute-force state vector calculations on 65 qubits for these circuits. We see that averaging over a circuit ensemble with Haar random two-qubit gates decreases the difficulty compared to ensembles which fix the two qubit gate and only average over single qubit gates, but not sufficiently to make the problem tractable. In addition, we show comparisons between two different basis choices for the sampling; the 24-dimensional permutation basis described above, labeled B_1 in the figure, and a basis using just the 14 linearly independent permutations which appear most frequently in the trajectories, labeled B_2, which we show performs much worse for the Haar ensemble and similarly for the iSwap-like gates.

4. *Cost estimate of exact tensor network simulation of $OTOC^{(2)}$*

The results in this section form the basis for the classical cost estimates of exact simulation that were presented in the main text. They are also used in conjunction with a gate-removal protocol in Section III C 5 to develop heuristic tensor-network-based methods for $\mathrm{OTOC}^{(2)}$ simulation.

Classical cost estimate of tensor networks for exact $\mathrm{OTOC}^{(2)}$ simulation — We observe that choosing the right basis to perform mid-circuit projections in TN contraction (as in Eq. (20)) greatly affects the variance of sampling [1]. In particular, we (i) choose the basis orthogonal to the initial product state basis (in our case, X basis orthogonal to the Z basis).

Alternatively, one could (ii) choose in a basis induced by the last mid-circuit layer of single-qubit gates, hence simplifying the last layer of two-qubit gates and thus reducing the cost of contracting the tensor network.

To make sure we choose the best projection basis, we compare (i) and (ii) in small systems. Because of the rejection sampling algorithm, the SNR follows:

$$\mathrm{SNR}_{\mathrm{projs}} = a_{\mathrm{SNR}}\sqrt{N_{\mathrm{projs}}} + b_{\mathrm{SNR}}. \tag{54}$$

Fig. S35 shows the coefficients a_{SNR} and b_{SNR} for both bases. The results are obtained by averaging SNR fits over multiple circuit geometries (with different depth and number of butterflies), with the shaded areas corresponding to the $25\% - 75\%$ confidence interval.

By scaling the coefficient a_{SNR} and b_{SNR} with the number of total qubits, we can estimate the expected SNR for larger system sizes. For the case of 65 qubits with 20 qubits open and a single projection, we estimate (i) an $\mathrm{SNR} = 16.67$ ($1.55 - 21.60$ confidence interval) using the X basis, and (ii) an $\mathrm{SNR} = 0.83$ ($0.73 - 0.87$ confidence interval) using the adversarial basis. Therefore, we further work in the X basis.

Fig. S34 (right panel) shows the expected time to contract a single 65-qubit $\mathrm{OTOC}^{(2)}$ tensor network with either no memory constraints or with memory constraints and 20 open qubits. The Frontier time is calculated by assuming a peak of $1.71{\cdot}10^{18}$ EXAFLOPS, with a 20% of efficiency [1]. The contraction path optimization has been performed on 20 `c2-standard-60` Google Cloud nodes (see Fig. S34).

Due to the contraction path structure (a single large tensor with many small tensors sequentially applied to it), the overhead of slicing is only $\sim 5\times$ larger than without any memory constraints. Besides, caching becomes less relevant for a sufficiently large number of gates removed. Finally, we compare `TNCO` to `Cotengra`, one of the widely used contraction path optimizers [19].

5. *Reducing tensor network costs via gate removal*

It is known that one needs to only simulate the effective volume of a circuit in order to determine output from a given qubit [9, 33, 40]. Claims comparing quantum and classical performance thus need to allow the classical computer to remove gates, or entire qubits. Qubit removal is typically impossible for OTOC and $\mathcal{C}^{(4)}_{\mathrm{off-diag}}$ circuits, as the long-distance nature of the signal allows the entire chip to be entangled. However, gate removal is possible; for example, the gates outside the intersection of the butterfly and measurement light cones may be trivially removed (and for this work already are excluded from the gate counts N_{2D} reported in the main text). Removing any other gate comes with a potential systematic error, which can only be determined in exact simulation, or extrapolation till convergence. On the beyond-classical boundary, both options become challenging.

In this section, we outline an approximate method to remove gates from a circuit based on the Monte-Carlo approximation. This routine correctly determines the order in which the gates should be removed, however

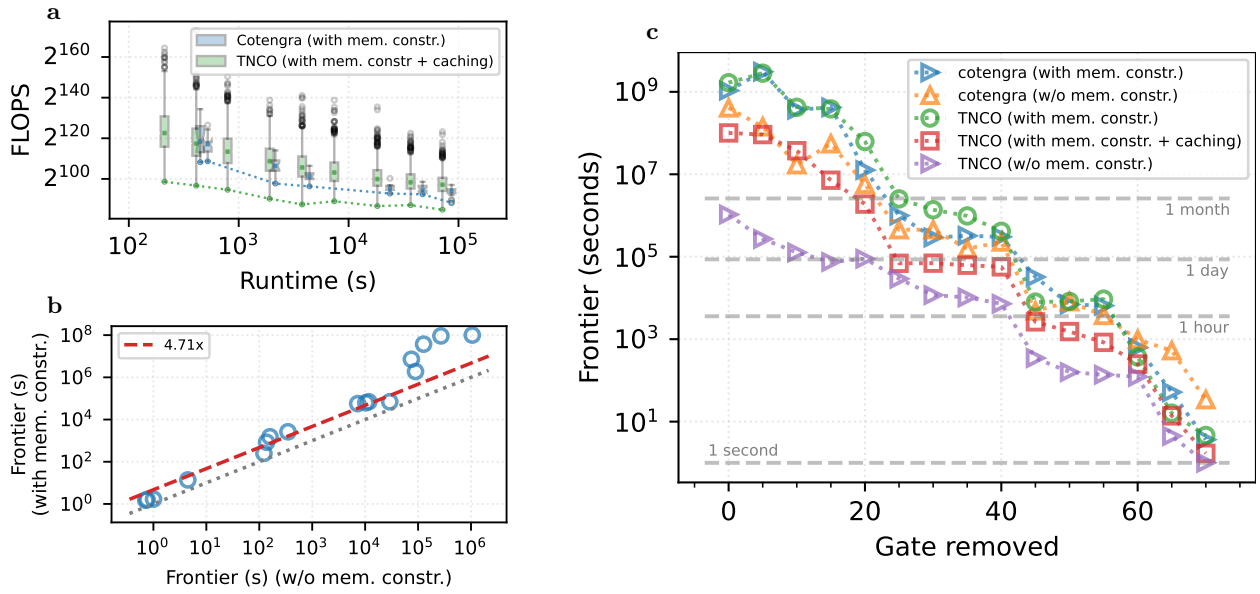


FIG. S34. **Optimization using tensor network contraction of the 65-qubit OTOC$^{(2)}$ circuits.** **a**, Total time spent to optimize the tensor network contraction of the 65-qubit OTOC$^{(2)}$ circuits with $N_{2D} = 1020$ on 20 `c2-standard-60` nodes on Google cloud, for zero gates removed. **b**, Overhead by imposing memory constraints (each point correspond to a different circuit with a given number of gates removed). For both plots, a maximum width of 33 and 20 open qubits are assumed. **c**, Estimated time on Frontier for different numbers of gates removed.

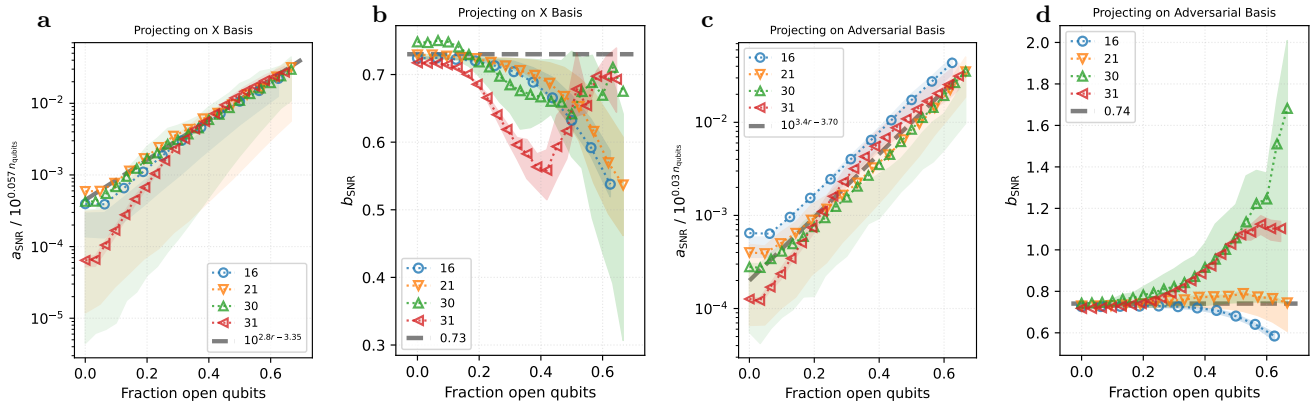


FIG. S35. **Scaling coefficient for SNR $= a_{\mathrm{SNR}}\sqrt{N_{\mathrm{projs}}} + b_{\mathrm{SNR}}$, with N_{projs} being the number of projections**. **a-b**, scaling for projecting on the X basis. **c-d**, scaling for projecting on the the adversarial basis. The expected SNR for 65 qubits and 20 open qubits are respectively 16.67 and 0.83. In all plots, the shared area corresponds to the $25\% - 75\%$ confidence interval.

the estimates of the systematic error of the gate removal can differ by an order of magnitude when compared to the exact results. As such, we present a *range* of possible estimates for competing classical algorithms, as is reported in the main text.

Heuristic classical algorithm for gate removal — Our algorithm proceeds by first ordering the gates by the reduction in SNR that results from their removal, and then trimming them one by one. [7] We estimate the SNR from the real circuit using a proxy classical simulation. Throughout this section and further, when a gate is removed, it is removed in all copies of U and $U^\dagger$.

We test the gate removal using the vanilla Monte

[7] In principle, this may not remove the optimal group of G gates, but testing all possible combinations is exponentially hard in G.

Gates Removed	SNR (MC)	$\log_2$(FLOPS)	Frontier Time	$\log_2$(FLOPS)*	Frontier Time*
0	∞	78.25	12 days, 5 hours	84.83	3 years, 2 months
5	61.17	76.29	3 days, 3 hours	84.70	2 years, 10 months
10	55.20	75.20	1 day, 11 hours	83.40	1 year, 2 months
15	39.83	74.43	20 hours, 45 minutes	81.02	2 months, 22 days
20	35.45	74.69	1 day, 50 minutes	79.09	21 days, 21 hours
25	24.36	73.08	8 hours, 6 minutes	74.32	19 hours, 15 minutes
30	20.26	71.76	3 hours, 15 minutes	74.33	19 hours, 24 minutes
35	18.72	71.57	2 hours, 51 minutes	74.17	17 hours, 20 minutes
40	15.84	71.06	2 hours	74.04	15 hours, 46 minutes
45	12.38	66.69	5 minutes, 48 seconds	69.60	44 minutes
50	10.39	65.56	2 minutes, 39 seconds	68.82	25 minutes
55	8.78	65.34	2 minutes, 17 seconds	67.95	14 minutes
60	7.88	65.18	2 minutes, 2 seconds	66.21	4 minutes
65	7.65	60.40	4.47 seconds	62.03	13.78 seconds
70	4.19	58.25	1.01 seconds	59.01	1.70 seconds

TABLE SIV. **Tensor network contraction estimates for the 65-qubit OTOC$^{(2)}$ circuits with $N_{2D} = 1020$ by removing gates that minimize the SNR reduction.** Columns with an asterisk indicates contractions with a tensor width limited to 33 and 20 open qubits. Frontier time is computed assuming a theoretical maximum of 1.71×10^{18} FLOPS, with a 20% of efficiency.

Carlo estimation (Section III B 1) of an OTOC$^{(1)}$ with the equivalent circuit U (half of the initial OTOC$^{(2)}$ circuit). We benchmark using the Monte Carlo gate ranking in this OTOC$^{(1)}$ circuit as a proxy for the gate ranking in the ideal $\mathcal{C}^{(4)}_{\text{off-diag}}$ circuits (with subtracted ABBA "classical" part) by simulating (i) $\mathcal{C}^{(4)}_{\text{off-diag}}$ in the 31-qubit circuits with 18 cycles (50 instances) exactly and (ii) OTOC$^{(1)}$ within Monte Carlo and computing the SNRs of each individual gate removal. The comparison is shown in Fig. S36a. If both the gate removal order and cumulative removal SNR are computed within (ii), the SNR is highly overestimated, as compared to the (i) case. For instance, at the point where SNR_{MC} crosses 6 (24 removed gates), $\text{SNR}_{\text{qsim}} = 2$, and in Fig. S36b-c we show the comparisons between the ideal (without removed gates) and the trimmed (24 removed gates) cases for (i) and (ii). While the Monte Carlo (ii) shows high agreement, the true signal (i) misses peaks, leading to the SNR of 2. We note that (i) uses the perfect gate ranking, but perhaps a non-optimal greedy cumulative removal starting with most insignificant gates. Finding the optimal strategy, however, would be exponentially hard in N_{removed}. Despite the perfect gate ranking used in (i), the SNR of $\mathcal{C}^{(4)}_{\text{off-diag}}$ is significantly lower than the Monte Carlo SNR obtained in (ii). Therefore, the SNR obtained in (ii) provides an upper bound on the desired SNR obtained in (i). In Fig. S37, we show the SNR ratios between (i) and (ii) for various system sizes and geometries to elaborate on the possible rescaling factor between the Monte Carlo proxy and the exact result.

Additionally, in Fig. S36d we look into the removal of individual gates. We observe that the Monte Carlo on average overestimates the SNR at the level of removal of individual gates, while the gate rankings are highly-correlated with the Spearman's rank correlation coefficient $r_s = 0.917$, which justifies using (ii) the gate ranking while trimming gates in the 65-qubit circuits.

Application to OTOC$^{(2)}$ circuits considered in this work — We now turn to estimating the complexity of simulating the 1020-gate data in Fig. 4f of the main text via tensor network methods, targeting the same SNR ~ 3.5 that was experimentally achieved. In Fig. S36e we plot the Monte-Carlo SNR estimates for this circuit with different numbers of gates trimmed, and observe that this follows a rough exponential decay law. Following this ordering, it remains to estimate the two systematic biases described above; the bias from the truncation itself, and the residual sampling noise from the finite number of tensor network projections Eq. (20).

Combining the results from Fig. S37 (which shows the range of estimates for the multiplier between the Monte Carlo proxy SNR and our true SNR), and the results from Fig. S35, we can extrapolate the optimal number of removed gates and number of projections to reach a target SNR (see Fig. S38). We assume that the error

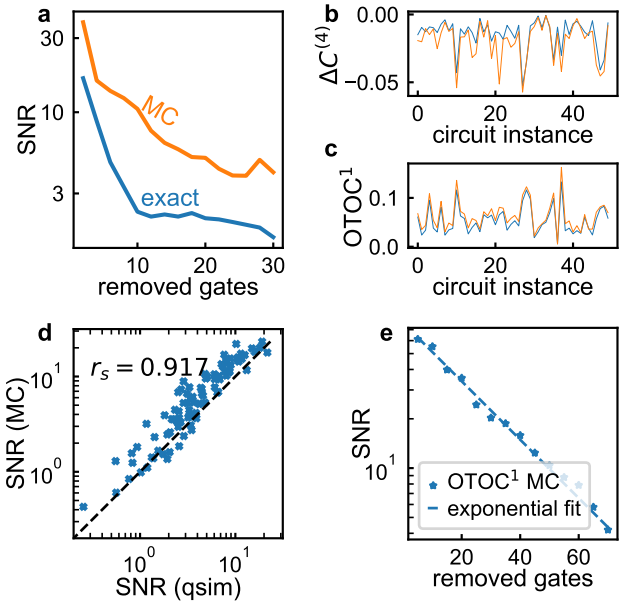


FIG. S36. **Monte Carlo-based gate removal.** **a,** In 50 instances of the 31-qubit circuit with 18 cycles, we computed SNR after removing each gate within the (i) exact simulation of $\mathcal{C}^{(4)}_{\mathrm{off-diag}}$ (qsim) and (ii) Monte Carlo simulation of OTOC^1 (MC). We ordered gates by their SNR impact increase, and show the SNR of cumulative removal. **b-c,** At the point where $\mathrm{SNR}_{\mathrm{MC}} = 6$ (24 removed gates), $\mathrm{SNR}_{\mathrm{qsim}} = 2$, and we compare of the signal without gate removal (blue) and with 24 gates removed (orange) for $\mathcal{C}^{(4)}_{\mathrm{off-diag}}$ (qsim, **b**) and $\mathrm{OTOC}^{(1)}$ (MC, **c**). **d,** The Spearman's rank correlation coefficient $r_s = 0.917$ indicates a high degree of correlation between (i) the ideal gate removal ordering and (ii) the Monte Carlo proxy. **e,** The Monte Carlo proxy SNR as a function of the number of removed gates (ordered using the Monte Carlo ranking) in the 1020 $\mathrm{OTOC}^{(2)}$ circuit. The exponential fit yields $\mathrm{SNR}_{\mathrm{MC}}(N_{\mathrm{removed}}) \approx 74.0\, e^{-N_{\mathrm{removed}}/24.4}$.

induced by removing gates is uncorrelated with the error induced by removing gates. Fig. S38 (left) shows the combined SNR by assuming a single projection. In the plot, we also assume that the multiplicative scaling factor between Monte Carlo SNR and exact SNR is 3.6 (2.1 — 6.5 confidence interval of $25\% - 75\%$), as well as an SNR of 16.67 for a single projection (see Section III C 4). We combine these two estimates in Fig. S38 (right), showing the expected time to contract a single tensor network on Frontier for a given target SNR. The runtime on Frontier is computed by dividing the FLOPS for the best tensor network contraction path we found using `TNCO` by the theoretical peak performance of Frontier ($1.71 \cdot 10^{18}$ FLOPS), assuming a 20% FLOPS efficiency [1].

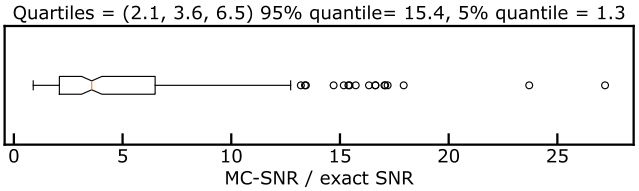


FIG. S37. **Box plot of ratio between Monte Carlo SNR and exact SNR.** Positions of various quantiles are given in the title text.

6. *Pauli-weight truncation algorithms*

Another possible approach to the $\mathrm{OTOC}^{(2)}$ computation is the *truncation in the Pauli space* [10, 11]. More specifically, one time-evolves the butterfly operator $B(t)$ in the Pauli space in sparse representation as in Eq. (43), but keeps only the Pauli strings with a limited number of non-identity entries, that is X, Y or Z. Importantly, unlike the applications considered in Refs. [10, 11], $\mathrm{OTOC}^{(2)}$ contains four copies of the butterfly operator. This makes it difficult to pre-select non-zero Pauli contributions based only on their Pauli weight L. In this section, we will argue that this approach is not applicable to the computation of higher-order OTOCs.

Large-weight Pauli contributions to $\mathrm{OTOC}^{(k)}$ — In contrast to expectation values and correlation functions where low-weight Pauli strings provide the dominant contribution, in the case of circuit-to-circuit variance of $\mathrm{OTOC}^{(k)}$ low weight Pauli strings do not contribute. Indeed a vanishing commutator $[B(t), M] = 0$ implies $\mathrm{OTOC}^{(k)} = 1$ and zero circuit-to-circuit variance. Whereas non-vanishing commutator $[B(t), M] \neq 0$ requires the expansion of $B(t)$ into Pauli strings to contain strings of sufficient weight to span the distance between B and M in space, corresponding to large weights for the geometries we consider. The distribution of Pauli strings over their Hamming weights is determined by the physics of operator spreading. For a random circuit a typical string has at each qubit operator I, X, Y, Z with roughly equal probability. Therefore for a system of n qubits we expect a typical Pauli weight of κn, where $\kappa = \frac{3}{4}$ for OTOC. Below we provide a more accurate estimate specific to $\mathrm{OTOC}^{(2)}$. Including this large Pauli weight makes Pauli truncation algorithms impractical for $\mathrm{OTOC}^{(k)}$.

Counting of nonzero contributions for $\mathrm{OTOC}^{(2)}$

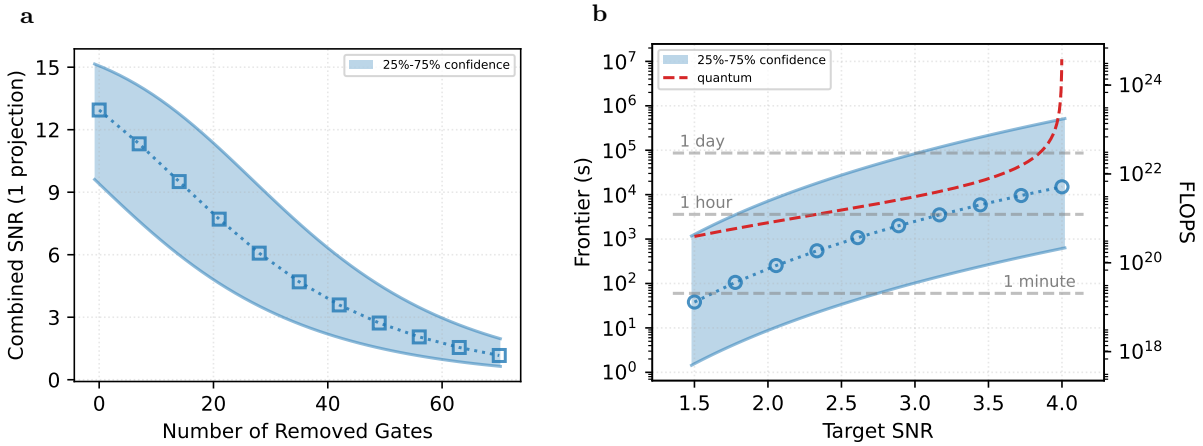


FIG. S38. **SNR estimates from tensor network contraction calculations, after removing gates.** **a**, Estimated SNR for the 65-qubit OTOC$^{(2)}$ circuits with $N_{2D} = 1020$ by removing gates, for a single projection. **b**, The runtime on Frontier for different target SNR. The dashed-line corresponds to the time to reach the same target SNR using the quantum device.

— To better understand why the pre-selection of non-zero Pauli contribution fails for OTOC$^{(2)}$, let us consider the following combinatorial argument. First, we estimate, Pauli strings of which weight L give the largest number of contributions $K(L)$ to the expectation. Here, $K(L)$ will be the total number of ways how one could pick four Pauli strings with Pauli weight L, such that they lead to a non-zero trace. For simplicity, we will assume that only four Pauli-strings with the same Pauli weight would satisfy the non-zero trace condition, which will serve as a lower bound.

In the sector of Pauli strings of the Pauli weight of exactly L, at each site, we will consider having I with the probability $1 - 3p = 1 - L/N$, and each of the X, Y and Z with the probability $p = L/(3N)$. Here, N is the number of sites. Sampling an operator at each of the N sites, we obtain

$$P(p) = \text{Prob}\left(\text{Tr}\left[P_1P_2P_3P_4\right] \neq 0\right) = \\ = 192p^4 - 192p^3 + 72p^2 - 12p + 1. \tag{55}$$

In total, there are $\binom{N}{L}3^L$ different Pauli strings with the weight L, which gives $\log K(L) = N \log P(p) + 4L \log 3 + 4 \log \binom{N}{L}$. The curve $\log K(L)$ for $N = 65$ (beyond-classical dataset) is shown in Fig. S39a. The curve peaks at $L^* = 47$. At this Pauli weight, there are $R(L^*) \sim 10^{38}$ possible Pauli strings.

Computation of OTOC$^{(2)}$ from the butterfly operator representation — Given the Pauli-weight truncated representation of the butterfly operator (with R terms), one requires to compute the OTOC$^{(2)}$ expectation. In the case higher-order OTOCs, the trace condition leaves numerous pairing options (64 in the case of OTOC$^{(2)}$) at each site. Therefore, we present an algorithm to directly compute OTOC$^{(2)}$ given some representation of B in the Pauli basis.

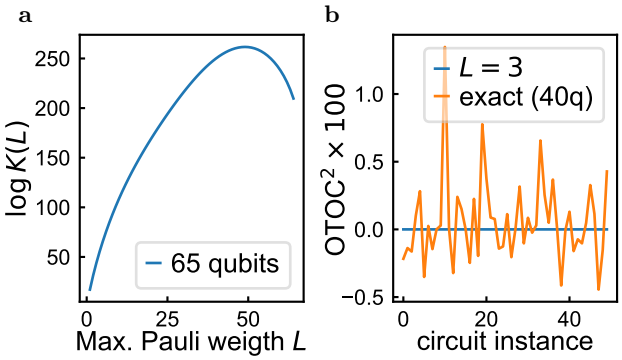


FIG. S39. **Simulation of OTOC$^{(2)}$ using Pauli weight truncation.** **a**, Combinatorial estimation of the number of contributions to OTOC$^{(2)}$ coming from each L-truncated Pauli sector. **b**, Application of the Pauli truncation algorithm to the 40-qubit OTOC$^{(2)}$ circuit shown in the main text.

For simplicity, we consider the $|0\rangle = |00\ldots0\rangle$ initial state which is an eigenstate of the measurement operator, and

$$\mathcal{C}^{(4)} = \langle 0|BMBMBMB|0\rangle. \tag{56}$$

First, we compute $|\psi\rangle = MB|0\rangle$ in $\mathcal{O}(R)$ time. Then, the expression reduces to

$$\mathcal{C}^{(4)} = \sum_{a,b} \psi^*(a)\psi(b)\langle a|BMB|b\rangle, \tag{57}$$

where the sums in a, b run over $\mathcal{O}(R)$ non-zero elements of the wave function $|\psi\rangle$. For each of the $\mathcal{O}(R^2)$ combi-

nations of a and b, we build $|\psi_a\rangle = B|a\rangle$, $|\psi_b\rangle = B|b\rangle$, and compute $\langle\psi_a|M|\psi_b\rangle$ in linear time. The total algorithm complexity is $\mathcal{O}(R^3)$.

Application in a realistic simulation — We apply the Pauli truncation algorithm to the 40-qubit OTOC$^{(2)}$ circuits shown in the main text. For $L = 3$, the maximum cache size is $R = 3^4 \times \binom{40}{3} = 0.8 \times 10^6$, and we can run the $\mathcal{O}(R^3)$ computation algorithm. For $L = 4$, with the cache size of $R = 23 \times 10^6$, the simulation becomes infeasible. The results for the $L = 3$ case are shown in Fig. S39b. For 40 qubits, we expect the most contributions to come from the sector with $L \sim 15$. Due to the excessive truncation, the result is vanishingly small. Likewise, renormalization of cache after each truncation leads to non-unitarity, and values of OTOC$^{(2)}$ uncorrelated to the ground truth.

Noisy OTOC$^{(k)}$ and Pauli-weight truncation — Noise in practical quantum processors provides an avenue to reduce the cost of Pauli path algorithms for correlation functions. Noise affects Pauli strings exponentially in their Hamming weight. Therefore there is a natural criteria for truncation of Pauli strings with the largest weights reducing the cost of the computation. This approach however does not reduce the cost of computing OTOC$^{(k)}$ provided the noise in the device is sufficiently weak. Indeed for an output of the Pauli truncation algorithm to demonstrate substantial correlation with the noise-free OTOC$^{(2)}$ Pauli strings of typical weights need to be included. For the circuit geometries we consider in this paper the distribution of Pauli weights is relatively narrow around its typical value. Therefore exploiting the effect of noise does not lead to a substantial reduction in the cost of Pauli truncation algorithm compared to its noise free version.

7. *Matrix product state methods*

In this section, we discuss the simulation of OTOC$^{(2)}$ large-scale circuits using matrix product state (MPS) and matrix product operator (MPO) methods. These methods' computational complexity is controlled by the entanglement generated in the time-evolution. At a cost of imperfect fidelity, these methods compress a wave function by truncating small Schmidt coefficients. At the same time, the generated entanglement could be decreased by the gate removal procedure (see Section III C 5), which also comes at the fidelity cost. In this section, we will demonstrate that the large entanglement generated in our experiments makes the application of such methods prohibitively expensive.

We first consider an MPS simulation. We enumerate the sites of a two-dimensional circuit in the *snake* manner, and form an MPS state

$$|\psi_\chi\rangle = U^\dagger B U M U^\dagger B U|0\rangle \tag{58}$$

we then truncate the bond dimension to χ after each gate. Lastly, we measure $\mathcal{C}_\chi^{(4)} = \langle\psi_\chi|M|\psi_\chi\rangle$.

We consider the MPS simulation cost of the beyond-classical OTOC$^{(2)}$ circuits in the two limiting regimes: (i) no truncation, but the entanglement is lowered by removing as many gates as possible, (ii) no gates removed, but the Schmidt values are truncated. In both these cases, we consider a minimum SNR ~ 3.

Entanglement in beyond-classical circuits — Determining the entanglement entropy for large quantum systems is a daunting task. Indeed, the complexity of computing the entanglement entropy by splitting a quantum system of n qubits in two is proportional to $\mathcal{O}(2^{3n/2})$. In some cases, bounds on the entanglement entropy can be computed efficiently by exploiting the fact that Clifford gates form a 2-design. As shown in [2, 9], if all 2-qubit gates are iSWAPs and all the 1-qubit gates are uniformly sampled from $Z^p\sqrt{X}Z^{-p}$, with $p \in \{-1, -1/4, -1/2, \ldots, 3/4\}$, then the entanglement entropy of such circuits is bounded by the entropy of the closely related Clifford circuits where p is sampled from $p \in \{-1, -1/2, 0, 1/2\}$ instead. More precisely, the purity $\overline{\mathrm{Trace}[\rho_L^2]}$ averaged over circuits, where ρ_L is the reduced density matrix after tracing half of the system, is the same for the two ensembles (see Section H5 of [2] for more details). Using the Jensen's inequality, it is possible to bound the entanglement entropy for non-Clifford circuits as

$$-\overline{\mathrm{Trace}[\rho_L \log_2 \rho_L]} \geq -\log_2 \overline{\mathrm{Trace}[\rho_L^2]}_{\mathrm{Clifford}}, \tag{59}$$

with $-\overline{\mathrm{Trace}[\rho_L^2]}_{\mathrm{Clifford}}$ being the average Rényi entropy using Clifford gates only.

Equation (59) is valid for the ensembles described above [2, 9], while the experimental circuits presented in this paper have a different gate ensemble. Nevertheless, small-scale numerical results in Fig. S42 show that the Clifford purity in Eq. (59) can be still used as an estimator for the experimental circuits. Indeed, Fig. S40

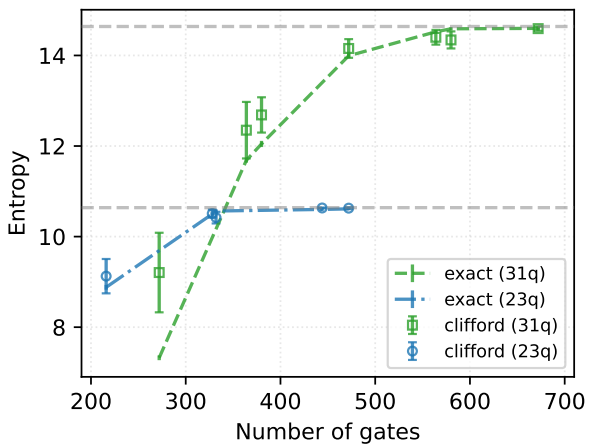


FIG. S40. **Entanglement entropy by varying the circuit depth for OTOC$^{(2)}$ with 23 qubits and 31 qubits.** The two sets of circuits use the same gate ensemble used experimentally. The horizontal gray-dashed lines correspond to the random Haar limit. The Rényi Clifford entropy well estimates the exact von Neumann entropy, including the saturation of the entropy at large depth.

compares the exact entanglement entropy to the Clifford Rényi entropy for OTOC$^{(2)}$ circuit with 23 and 31 qubits respectively, by varying the number of entangling gates. Both set of circuits use the same gate ensemble used in the experimental setup. More precisely, the figure shows that the Clifford Rényi entropy is a good estimator of the entanglement entropy, correctly predicting when the entanglement entropy saturates (indicated by the dashed-gray line). We also report the comparison by varying the number of removed gates in U, Fig. S41. The main observation is that the Clifford Rényi entropy is still a good estimator of the entanglement entropy even when gates are removed, as long as the final SNR is larger than the baseline $1/\sqrt{2}$.

Therefore, assuming that the Clifford estimates hold for larger systems, we expect an entanglement entropy of ~ 30 for the 65 qubits and $N_{2D} = 1020$ entangling gates (see Fig. S42).

Simulation cost without truncation — We begin with analyzing the case (i), where we decrease the maximum wave function entanglement entropy by removing gates from the circuit. As shown in Section III C 5, removing ~ 70 gates gives the SNR of at most 3. After such removal, as shown in Fig. S42, the Rényi entropy is ~ 21, which leads to the required bond dimension of $\chi_{\min} = 2^{21}$, and the respective FLOP count of at least $8\chi_{\min}^3 = 2^{66} = 7.3 \times 10^{19}$, assuming single complex precision. With the Frontier FLOPs of 2×10^{18} and assuming effective performance of 20 %, we obtain a single SVD run time of at least 3 minutes, making this approach not faster than the direct tensor network contraction.

Simulation cost with truncation — Finally, we consider the case where the truncation is the sole source of finite SNR. We first estimate the minimum bond dimension $\chi^*_{\mathrm{SNR}=3}$ required to reach SNR ~ 3, by linking $\chi^*_{\mathrm{SNR}=3}$ to the maximum Rényi entropy $S^{\max}_{\text{Rényi}}$ generated during the composition of a state Eq. (58). We do that by running MPS simulations in smaller systems with up to 30 qubits. The results are shown in Fig. S43. The panels Fig. S43a-e show SNR as a function of maximum bond dimension in the systems of various size and depth. Fig. S43f shows the minimum required bond dimension $\chi^*_{\mathrm{SNR}=3}$ as a function of the maximum Rényi entropy $S^{\max}_{\text{Rényi}}$. Fitting as an exponential gives

$$\chi^*_{\mathrm{SNR}=3} = 3.9\, e^{0.56\, S^{\max}_{\text{Rényi}}}, \tag{60}$$

where we note that larger systems' points mostly lie above the fit, which signals that the fit likely presents a conservative estimate. As shown in Fig. S42, the Rényi entropy in the beyond-classical OTOC$^{(2)}$ circuits without gate removal is ≈ 29, which gives $\chi^*_{\mathrm{SNR}=3} \sim 30 \times 10^6$, and therefore per-SVD 2.8×10^{23} FLOPS, resulting into 8 days on Frontier assuming ~ 20 % FLOPS efficiency, making this approach impractical.

OTOC$^{(2)}$ simulation in the MPO formalism — Finally, we test the prospects of an MPO simulation. We time-evolve the butterfly operator in the form of an MPO, and then compute (contract) an observable using several copies of the MPO. To obtain a reliable scaling, here we study a family of one-dimensional systems with the butterfly operator placed two steps behind the front with the butterfly velocity $v_B = 3/5$. The results are shown in Fig. S44, where we compare the cost of an MPO simulation to the cost of the direct state-vector simulation. For all considered system sizes, a state-vector simulation requires less computation. Therefore, due to the large operator entanglement entropy, the MPO approach does not give advantage over the direct state-vector simulation of OTOC$^{(2)}$, which is impractical in the large-scale circuits.

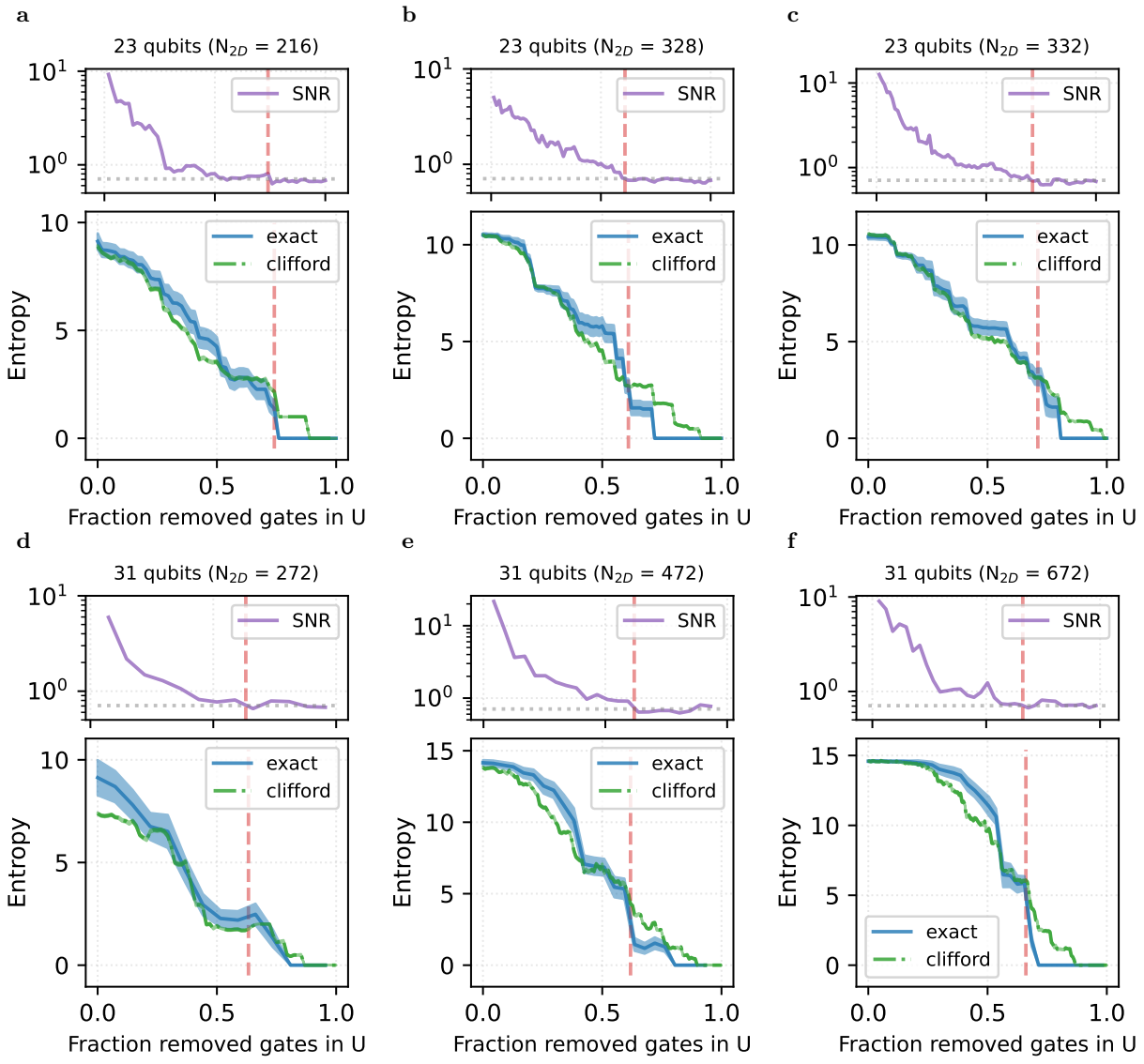


FIG. S41. **Comparison between the exact von Neumann entropy and the Rényi Clifford entropy for OTOC$^{(2)}$ circuits with 23 qubits (a-c) and 31 qubits (d-f).** The two sets of circuits use the same gate ensemble used experimentally. The horizontal gray-dotted line corresponds to the baseline SNR of $1/\sqrt{2}$, while the horizontal red-dashed line corresponds to the number of removed gates when the SNR drops below the baseline. As long as the SNR is larger than $1/\sqrt{2}$ the Rényi Clifford entropy is a good estimator of the exact von Neumann entropy.

8. *Neural quantum states*

In this section, we discuss the prospect of applying the *Neural Quantum States* (NQS) method [3], wherein a quantum state is variationally represented in form of a neural network, $\Psi(\boldsymbol{\theta}, s) \to \psi(s)$, a function of trained parameters $\boldsymbol{\theta}$, which, given a bit string s, produces the respective element of the wave function $\psi(s)$. This approach has been applied to time-evolution in cases of either adiabatic evolution [41–43] or special regimes of the transverse-field Ising model [44–47].

In this section, we fit an exact state-vector as a neural network with a variable number of parameters. This allows us to assess the *expressivity* of the NQS ansatz without delving into particular details of the time-evolution implementation. At first, we use a standard architecture with $N_h = 2, 3, 4$ dense hidden layers and `ReLu` nonlinearities, which has two outputs predicting $\log|\psi(s)|$ and $\arg\psi(s)$ [4]. This choice of ansatz is governed by the random circuit nature of time-evolution (as opposed to the Hamiltonian evolution) and the absence of any spatial symmetries, which disfavors convolutional neu-

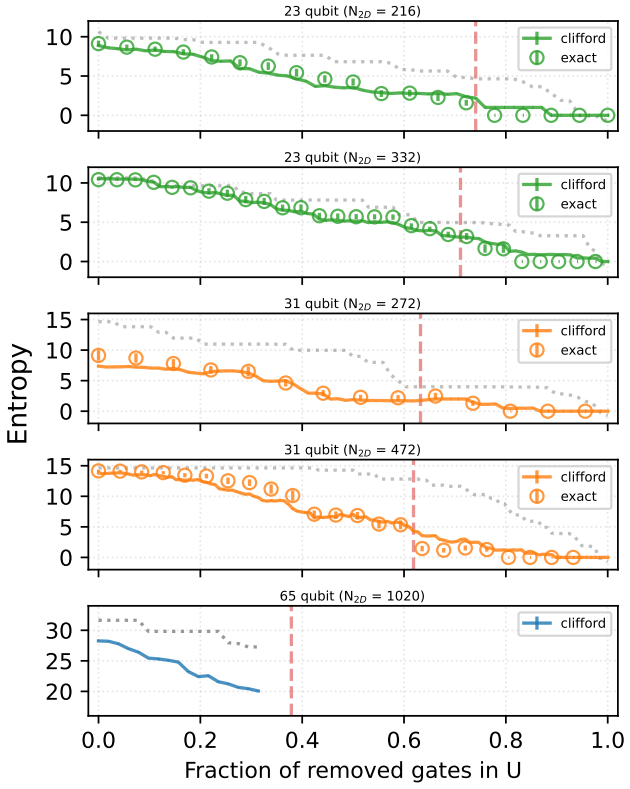


FIG. S42. **Entanglement entropy for OTOC$^{(2)}$ circuits of different system sizes.** The plot compares the exact von Neumann entropy $-\overline{\text{Trace}[\rho_L \log_2 \rho_L]}$ compared to the Rényi entropy $-\log_2 \overline{\text{Trace}[\rho_L^2]}_{\text{Clifford}}$. Vertical dashed-red lines correspond to the fraction of number of gates in U for which the SNR drops below the baseline of $1/\sqrt{2}$, while dotted-gray lines correspond to the maximum entanglement entropy for random Haar states. The SNR limit for 65 qubit is computed by using estimates from cached Monte Carlo. For 65-qubit circuit with $N_{2D} = 1020$, 70 removed gates correspond to the fraction $(4 \times 70)/1020 \approx 0.274$.

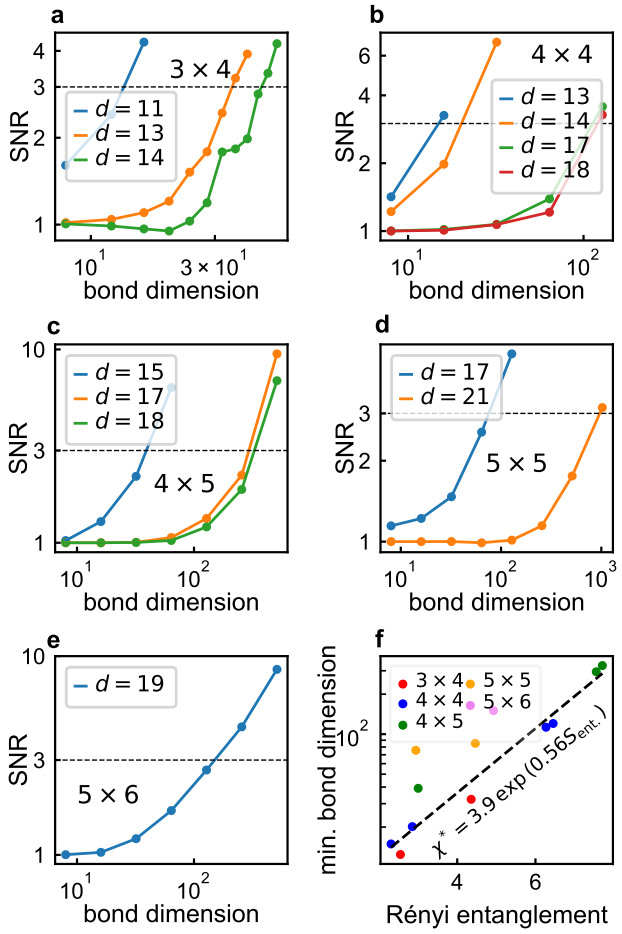


FIG. S43. **MPS simulation of OTOC$^{(2)}$ circuits of size $l_x \times l_y$. a-e,** SNR as a function of the maximum bond dimension for several depths. We extract $\chi^*_{\text{SNR}=3}$ using a polynomial extrapolation. **Fitting, f** Fitting $\chi^*_{\text{SNR}=3}$ as a function of $S^{\max}_{\text{Rényi}}$ using an exponential fit.

ral networks (CNNs). Later, we consider CNN and the ansatz with complex weights and biases, which avoids the amplitude-phase separation.

In Fig. S45, we present the case $N_h = 2$, and we fit the final state-vector directly. We consider a family of 31-qubit circuits used in the main text, and control their size and entanglement by truncating the depth at a particular value. We optimize the neural network parameters $\boldsymbol{\theta}$ to obtain the maximum possible overlap with the exact wave function using the ADAM optimizer [49] (the variations, such as AdaBound [50] yield practically similar results). We normalize the x–axis for all curves by the maximum Hilbert space size of a particular system. To obtain the best possible fit, we train a neural network starting from 10 different random initial conditions.

Fig. S45 shows that the curves feature near-"universal" behavior and have an inflection point around the Hilbert space size, $N_{\text{params}} \sim 0.1|\mathcal{H}|$, with fidelity ~ 0.5, slowly decreasing with the system size. If we assume that the fidelity F contributes to the signal, while the infidelity $1-F$ contributes to the noise, we can approximately write SNR $\propto F/(1-F)$, which we illustrate in Fig. S46a. This means that achieving the experimental SNR ~ 3 would require to train a neural network with the parameters count of order the full Hilbert space size, which we deem impractical.

This result is expected since, as Section III C 7 shows, the wave functions produced in our OTOC$^{(2)}$ circuits have entanglement close to the maximum (with the Rényi

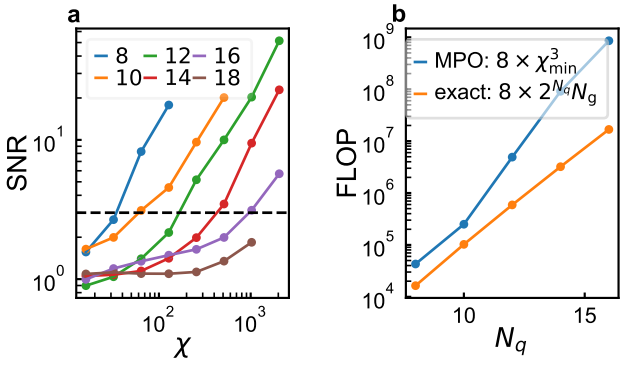


FIG. S44. **MPO simulation of one-dimensional OTOC$^{(2)}$ circuits. SNR as a function of the maximum bond dimension a,** In one-dimensional geometries with $N_q = 8, 10, \ldots, 18$, MPO simulation SNR as a function of the maximum virtual MPO bond dimension χ. **FLOP extrapolation b,** We compare the lower bound on the MPO simulation complexity $8\chi^3_{\min}$ to the cost of a state-vector simulation, $8 \times 2^{N_q} \times N_{\text{gates}}$.

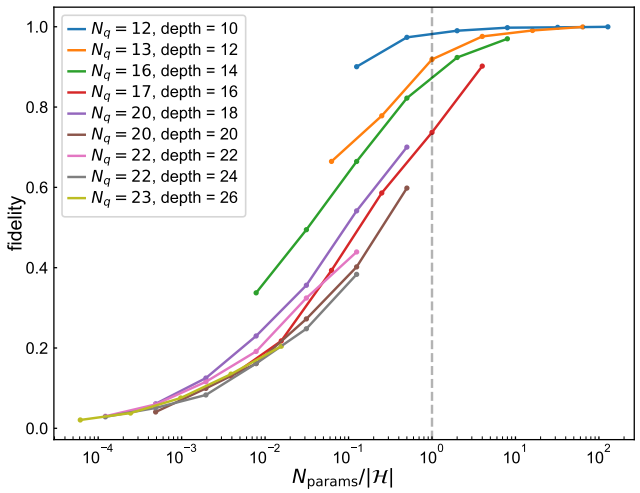


FIG. S45. **Neural Quantum States (NQS) applied to a typical OTOC$^{(2)}$ wave function used in this work.** To obtain each point, we select the best run of ten random initializations, and average all curves over 5 random circuit instances. The curves show that the overlap between the NQS wave function and the state-vector (exact) wave function, as a function of the number of parameters in the neural network, divided by the Hilbert space size.

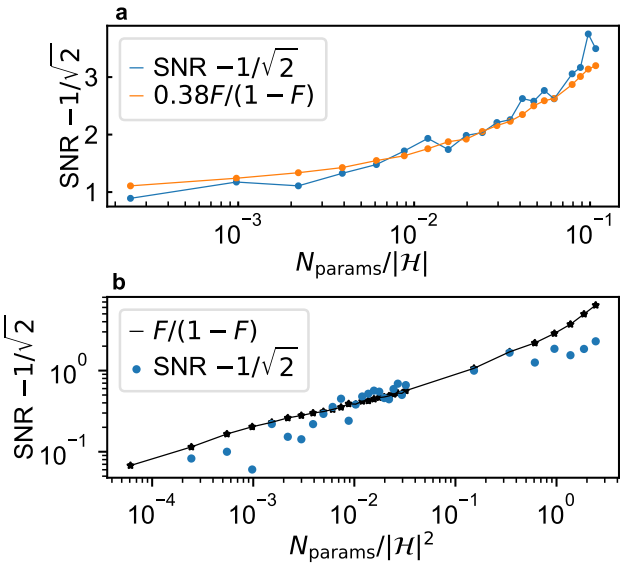


FIG. S46. **Fidelity and SNR comparison. a,** In a 16-qubit one-dimensional system with 36 layers of random $SU(4)$ two-qubit gates, we plot the SNR deviation from benchmark $\text{SNR} - 1/\sqrt{2}$ (computed over 30 random circuit instances) and $\alpha F/(1-F)$, where we fit $\alpha \sim 0.38$, as a function of the parameter count. We observe that a large SNR in the OTOC$^{(2)}$ observable is only achievable at a significant fidelity. **b,** In a 16-qubit one-dimensional chain, in the OTOC$^{(1)}$ circuits, we use NQS in the Pauli basis to fit the time-evolved $B(t)$ operator and plot $F/(1-F)$ (where F is the fidelity of the operator representation). We also plot the SNR deviation from the benchmark $\text{SNR} - 1/\sqrt{2}$ (computed over 30 random circuit instances).

entropy being $\sim$90 % of the maximum, see Fig. S42), and are therefore highly chaotic. While, in a few cases, it is possible to find a polynomial ansatz to represent a particular volume-law entangled wave function of a ground state [51–54], in the general case of a time-evolved wave function constructing such parametrization is practically hard and a typical NN architecture would require a an exponential number of parameters [5, 55], see also [54]. Therefore, unlike the non-generic cases of low-entangled or low-energy wave functions, their accurate representation, in practice, requires a number of parameters of the order of $|\mathcal{H}|$.

The most prominent example of successful application of the NQS method to the time-evolution is the transverse-field Ising model initiated in the product state polarized along the transverse field [44–47] for different field strengths $h = 2h_c$, h_c, $0.1h_c$ (here, $h_c \approx 3.044$ is the critical value). Importantly, the evolution for $h = 2h_c$ and $h = h_c$ reaches only ~ 0.08 and < 0.2 of the maximum von Neumann entanglement entropy, respectively. Hence, these wave functions are not generic and an efficient compression in terms of either neural networks, or matrix-product states was found. In particular, the case

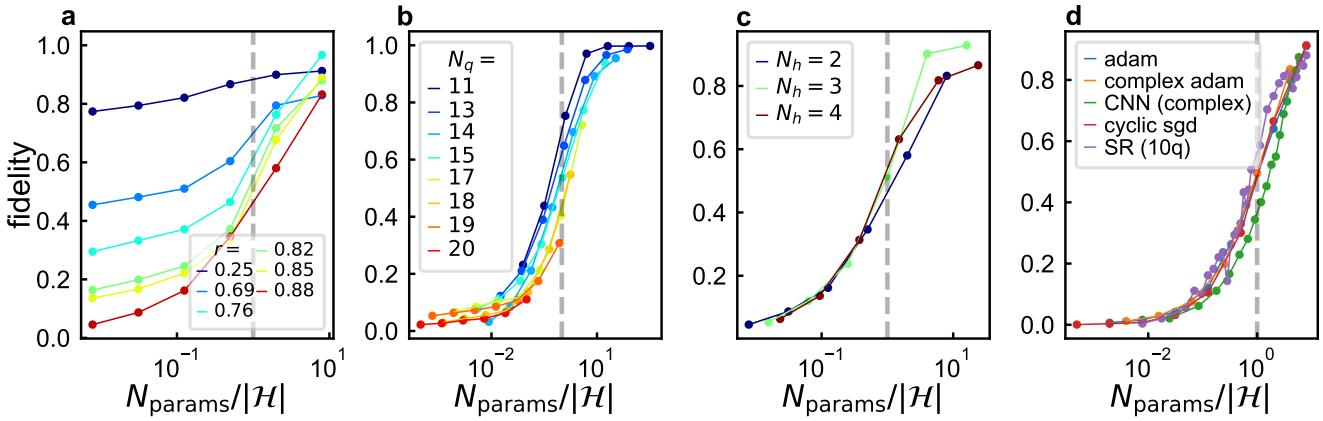


FIG. S47. **NQS expressivity tests in various scenarios a,** In a 16-qubit one-dimensional chain with $SU(4)$ random two-qubit gates at various depths, we plot fidelity (best in 10 runs) as a function of $N_{\text{params}}/|\mathcal{H}|$ (two hidden layers). Here, the relative entropy is defined as $r = S^{\text{ent.}}_{\text{Rényi}}/\max S^{\text{ent.}}_{\text{Rényi}}$. **b,** For one-dimensional chains with N_q qubits, we plot fidelity (best in 10 runs) as a function of $N_{\text{params}}/|\mathcal{H}|$ (two hidden layers). For each system size, we choose the depth such as $0.85 \leqslant r \leqslant 0.90$, in order to replicate the case for beyond-classical dataset. **c,** In a 16-qubit chain and depth 70, comparison of architectures with $N_h = 2, 3, 4$ hidden layers. **d,** In a 16-qubit chain at depth 48, we contrast using the Adam optimizer, stochastic gradient descent with Nesterov momentum, cyclic learning rate and stochastic reconfiguration (SR) [48], which mimics imaginary-time evolution of the wave function. We also attempt to make all the parameters of the neural network complex to avoid the amplitude-phase separation and predict the entries of the wave function directly (complex Adam) and do the same in a CNN (complex). All experiments collapse on the same curve.

$h = 0.1h_c$ is near-classical and, therefore, it allows for a near-exact projected entangled pair state description with a very small bond dimension of $D = 2$ [56], which a neural network can learn as well. It is also important to notice that MPS simulations may provide a better accuracy than NQS simulations for transverse-field Ising models at the finite energy density phase transition that require a large bond dimension, as shown in [57].

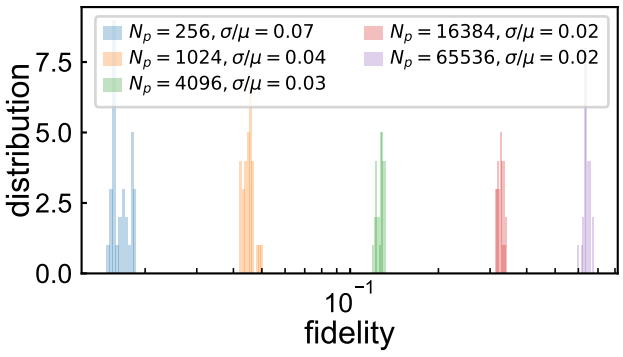


FIG. S48. **Distributions of fidelities.** Here, we vary the number of hidden features and launch 30 NQS optimizations starting from different initial random parameter initializations. We observe low variance-to-mean $\sigma/\mu \ll 1$ ratios.

By varying the circuit depth and thus tuning the entanglement, we observe a potential reduction of the number of parameters required by the NQS simulations, which is substantial in the low-entanglement regime, and disappears once the wave function becomes generic enough, as we show in Fig. S47a. However, if we focus on the depths where the Rényi entropy is over 85 % of the maximum, the inflection point is always at around $N_{\text{params}} \sim |\mathcal{H}|$ (see Fig. S47b), signalling that our beyond-classical OTOC$^{(2)}$ wave functions can not be efficiently expressed in terms of a NQS.

For the case of OTOC$^{(1)}$, we check if NQS is capable of efficiently compressing the time-evolved butterfly operator $B(t)$ in the Pauli basis, instead of the state in the computational basis, probing the *operator entanglement*. The result is shown in Fig. S46b. We observe that obtaining an accurate $B(t)$ representation requires achieving substantial fidelity, which is even more difficult in the Pauli basis. This agrees well with the fact that cached Monte Carlo simulations (sec. III B 2) require numerous projections, and therefore the $B(t)$ operator evolution has large *quantum magic*, hindering the efficient compression of $B(t)$ with NQS [58, 59].

We now check if other neural network architectures can

represent the typical OTOC wave functions using a polynomial number of parameters. First, we tested the effect of varying the number of hidden NQS layers, with results shown in Fig. S47c. We clearly observe that the representation quality depends only on the number of parameters, N_{params}. The use of a convolutional neural network did not improve the scaling, which is expected due to the absence of any spatial symmetry, as shown in Fig. S47d. Furthermore, avoiding the amplitude-phase separation and considering a neural network with complex parameters does not improve the scaling either (Fig. S47d).

Finally, we discuss the potential effect of local minima in the optimization. Fig. S47d shows no practical difference between choosing the Adam or stochastic gradient descent with Nesterov momentum ($\eta = 0.9$), stochastic reconfiguration [48] and cyclic learning rate, which is believed to avoid sharp local minima better than Adam [60]. Furthermore, we consider the distribution of fidelities in multiple optimizations starting from different random initializations in Fig. S48. We observe very small variance-to-mean ratio, i. e. small dependence of the outcome on the initial network initialization.

In conclusion, while not a formal proof, our foregoing analysis indicates that an efficient implementation of NQS algorithms outperforming TN contraction for simulating generically chaotic circuits such as $\text{OTOC}^{(2)}$ has significant practical challenges. The results do not preclude more efficient NQS optimization routines than what we have attempted here, which we leave as a subject of future work.

IV. THEORY OF OTOCS AND HIGHER-ORDER OTOCS

In this section, we develop theoretical models of $\text{OTOC}^{(k)}$ that describe the generic physics underlying the experiments performed in the main text. The first is an analysis of Haar random circuit ensembles, Section IV A, using a mapping to a statistical theory derived by exact averages over the gate ensemble, as has been done previously for OTOC in Refs. [1, 29, 30, 61]. In this framework, we analyze circuit-to-circuit variance of OTOC using perturbation theory in $1/d$, where d is the Hilbert space dimension of a qudit, and verify it with a tensor network simulation for large 1D chains of up to 1000 qudits, see IV A. We demonstrate that the magnitude of the variance scales as a slow polynomial of inverse circuit depth. The theoretical and numerical prediction of OTOCs with $k > 1$ and their circuit-to-circuit variances for $k \geq 2$ is challenging as the respective statistical models suffer from a sign problem, Section III C 3. This complexity also increases with k. In Section IV A 3 we use tensor network simulations for large 1D chains to study the circuit average of $\text{OTOC}^{(2)}$, which demonstrates a time dependence consistent with the spectral gap closing phenomena discussed in the main text. For circuit-to-circuit variance of $\text{OTOC}^{(2)}$ we rely on exact numerical simulations up to $n = 36$ qubits ($d = 2$). We find that the circuit variance for $k = 2$ decays slowly with depth. Additionally, we numerically study the decay of $\mathcal{C}^{(2)}_{\text{off-diag}}$ and $\mathcal{C}^{(4)}_{\text{off-diag}}$, which are constructed as in the main text by subtracting from $\mathcal{C}^{(2)}$ or $\mathcal{C}^{(4)}$ respectively the average of the same over inserting Pauli strings midway through the circuit. For $\mathcal{C}^{(2)}$, the Pauli insertions induce pairings of the forward and backwards trajectory of the operator; nonetheless, we show that this captures in large 1D circuits most of the full signal, with the remainder $\mathcal{C}^{(2)}_{\text{off-diag}}$ approaching 0 polynomially but with a faster power than $\mathcal{C}^{(2)}$. For $\mathcal{C}^{(4)}$, two separate sets of uncorrelated Pauli insertions are introduced for the first and second copies of the evolution in the circuit. This allows capturing part of the $\mathcal{C}^{(4)}$ signal, but there is a sizable remainder due to contributions that satisfy the trace condition with pairing of trajectories, $\mathcal{C}^{(4)}_{\text{off-diag}}$. The decay of $\mathcal{C}^{(4)}_{\text{off-diag}}$ is large 1D circuits is quite slow, but our numerics don't suffice to resolve whether the asymptotic fraction of the full $\mathcal{C}^{(4)}$ signal that this represents saturates or drops to 0.

In Section IV B, we develop a theory of $\text{OTOC}^{(k)}$ for *all* k by relating these objects to the spectrum of the *correlation operator* (see the main text and Section IV B for the definition). First, we introduce an exactly-solvable random-matrix model that has no spatial structure but has a tunable parameter that controls the degree of scrambling. Variation of this parameter captures the evolution between the limits of no dynamics and of a fully scrambling dynamics. We show that the spectrum of the correlation operator undergoes a gap closing transition with the increase of the scrambling parameter; we analyze the properties of this transition and find its effect on $\text{OTOC}^{(k)}$ values. Additionally, we use exact numerics to argue that the picture of the spectral phase transition holds also in local circuits, with the gap closing as one tunes across the operator front.

The combination of the circuit models we analyze in this Section demonstrate a set of features generic for out-

of-equilibrium quantum dynamics: large circuit-specific signal OTOC$^{(k)}$ which can be efficiently measured on a quantum processor and absence of known efficient classical simulation algorithms for $k > 2$. This motivates circuit-specific OTOC$^{(k)}$ values as a resource for Hamiltonian learning with the potential for quantum advantage in practical settings.

A. Statistical theory of Haar averaged random circuits

In circuits with Haar random gate ensembles, averages over the Haar ensemble can be exactly mapped to "classical" statistical lattice models which can be described as S_k ferromagnets; see the review Ref. [62] for an overview of the many previous works involving these mappings, and Refs. [1, 29, 30, 61, 63] for previous applications to the study of OTOCs.

In this section we analyze OTOCs for qudits subject to a circuit $U(t)$ of depth t that consists of a bricklayer pattern of Haar random two-qubit gates, which take the form

$$\mathcal{C}^{(2k)}_{B,M}(t) = \frac{1}{d^n}\text{Trace}\left[(B(t)M)^{2k}\right] \tag{61}$$

for systems of n qudits of dimension d, where B and M are the butterfly and measurement operators respectively. We consider average values over the circuit ensemble, which we denote with the horizontal bar. We rewrite the previous equation as

$$\overline{\mathcal{C}^{(2k)}_{B,M}(t)} = \frac{1}{d^n}\text{Trace}\left[\pi_{(1,2,\dots k)}\overline{B(t)^{\otimes 2k}}M^{\otimes 2k}\right], \tag{62}$$

a correlation function in an enlarged Hilbert space that consists of $2k$ copies of the original. π_σ denotes a permutation operator that acts on the $2k$ copies of Hilbert space by sending the j-th copy to the $\sigma(j)$-th copy, where σ is a permutation from the symmetric group S_{2k}. We adopt the standard cycle notation for permutations [64] — thus, $(1, 2, \dots 2k)$ refers to the permutation that sends j to $(j + 1) \pmod{2k}$.

Additionally, we consider the variances

$$\Delta^{(2k)}_{B,M}(t) = \overline{\mathcal{C}^{(2k)}_{B,M}(t)\mathcal{C}^{(2k)}_{B,M}(t)} - \overline{\mathcal{C}^{(2k)}_{B,M}(t)}\;\overline{\mathcal{C}^{(2k)}_{B,M}(t)}. \tag{63}$$

The first term can be rewritten as above using a Hilbert space with $4k$ copies of the original,

$$\begin{aligned}&\overline{\mathcal{C}^{(2k)}_{B,M}(t)\mathcal{C}^{(2k)}_{B,M}(t)}\\ &= \frac{1}{d^{2n}}\text{Trace}\left[\pi_{(1,2,\dots 2k)(2k+1,\dots 4k)}\overline{B(t)^{\otimes 4k}}M^{\otimes 4k}\right].\end{aligned} \tag{64}$$

In the average expressions, each gate U_{ij} acting on a pair of sites i, j in the circuit appears in the average $r = 2k$ times as U_{ij} and $2k$ times as $U^\dagger_{ij}$; in the second moments, $r = 4k$ times each. Generally, in an expression involving r copies of U_{ij} and r copies of $U^\dagger_{ij}$ of a Haar random gate, the average

$$\mathcal{P}_r \equiv \overline{\bigotimes_{s=1}^{r} U_{i_s j_s}\left(U_{i'_s j'_s}\right)^\dagger} \tag{65}$$

can be evaluated exactly as a projection onto the space of Haar invariant operators. A convenient albeit non-orthogonal basis for the invariant subspace [65] is the set of permutation operators, σ, τ, permuting the n copies, $|\sigma\rangle = \prod_{s=1}^{r}\delta_{i_s,\sigma(j'_s)}$. We rewrite the circuit average $\mathcal{P}_r$ in this subspace as

$$\mathcal{P}_r = \sum_{\sigma,\tau}|\tau\tau\rangle\text{Wg}(\tau\sigma^{-1})\langle\sigma\sigma|,. \tag{66}$$

Here $\text{Wg}(\tau\sigma^{-1})$ is the Weingarten function corresponding to the two-qudit Hilbert space dimension d^2, which can be computed by inverting the overlap matrix of the permutations [65, 66]. Overlaps between permutations are given by $\langle\alpha|\beta\rangle = d^{r-|\alpha^{-1}\beta|}$, where $|\alpha^{-1}\beta|$ is the minimum number of elementary transpositions comprising the permutation $\alpha^{-1}\beta$. Dynamics is generated by applying two-qudit projectors $\mathcal{P}_r$, which by Eq. (66) has matrix elements

$$\mathcal{P}_r|\mu\nu\rangle = \sum_\tau T_{\tau\tau,\mu\nu}\,|\tau\tau\rangle\,, \tag{67}$$

$$T_{\tau\tau,\mu\nu} = \sum_\sigma \text{Wg}(\tau\sigma^{-1})\langle\sigma|\mu\rangle\langle\sigma|\nu\rangle. \tag{68}$$

Specifically, for each Haar random two-qudit gate between qudits at sites i, j, $\mathcal{P}_r$ is applied to sites i, j starting with r copies of the initial operator $B(t = 0)$.

The projectors force the operator dynamics to take place entirely within the subspace spanned by products of permutations on every site; thus we can write the average evolved operator $\overline{B(t)^{\otimes r}}$ in the form

$$\sum_{\sigma_i\in S_r} B_{\sigma_1,\dots,\sigma_n}(t)\,|\sigma_1,\sigma_2,\dots,\sigma_n\rangle \tag{69}$$

for $t \geq 1$. In this basis, uniform two-qudit operators are stationary $\mathcal{P}_r|\sigma\sigma\rangle = |\sigma\sigma\rangle$. Operators with globally constant permutation factors $|\sigma\sigma...\sigma\rangle$ are thus stationary states of the dynamics; for this reason these models are referred to as S_r ferromagnets [62, 67]. Non-trivial dynamics occurs only along domain walls between permutations.

Additionally, as a consequence of the unitarity of the circuits, these dynamics preserve $r!$ conserved charges

$$\langle \omega\omega ... \omega | B^{\otimes r}(t)\rangle \text{ for each } \omega \in S_r.$$

For reasons that will become clear when we analyze the dynamics in the following sections, we find it convenient to use a change of basis that is a simple rescaling of the permutations $\hat{\sigma} = \sigma/\langle\omega|\sigma\rangle$, where $\omega = (1, 2, \dots 2k)$ for the mean OTOC$^{(k)}$ calculation and $\omega = (1, 2, \dots 2k)(2k+1, 2k+2, \dots 4k)$ for the second moment OTOC$^{(k)}$ calculation. Quantities in this rescaled basis are decorated with hats below to distinguish them. This results in a equivalent dynamical rule

$$\mathcal{P}_r|\hat{\mu}\hat{\nu}\rangle = \sum_{\tau} \hat{T}_{\tau\tau,\mu\nu}\,|\hat{\tau}\hat{\tau}\rangle\,, \tag{70}$$

$$\hat{T}_{\tau\tau,\mu\nu} = \sum_{\sigma} \mathrm{Wg}(\tau\sigma^{-1}) \frac{\langle\sigma|\mu\rangle\langle\sigma|\nu\rangle\langle\omega|\tau\rangle^2}{\langle\omega|\mu\rangle\langle\omega|\nu\rangle}, \tag{71}$$

which by construction has the additional property that it conserves the total weight in the rescaled basis

$$\langle\omega\dots\omega| \sum_{\sigma_i\in S_r} \hat{B}_{\sigma_1\dots\sigma_N}(t)|\hat{\sigma}_1\dots\hat{\sigma}_n\rangle = \sum_{\sigma_i\in S_r} \hat{B}_{\sigma_1\dots\sigma_n}(t).$$

Below, we analyze how the domain walls evolve in the calculations for mean OTOC$^{(1)}$, second moment of OTOC$^{(1)}$, and mean OTOC$^{(2)}$. We attack the problem both perturbatively in $1/d$ and use tensor network algorithms to directly compute the dynamics of the permutation wavefunction Eq. (69) for various values of d. These results are used to validate the perturbative results and show how they extend to small values of d.

1. *Circuit average OTOC$^{(1)}$*

A precise quantitative description of dynamics of the mean OTOC$^{(k)}$ for $k = 1$ is well known [1, 29, 61]; the circuit average dynamics is Markovian and can be described as a stochastic growth of a domain. We review the key features of these results here as they are relevant for understanding the following sections.

For simplicity, we will analyze the case where the butterfly operator B is chosen as a non-identity Pauli operator on the left-most site; after averaging the first Haar random gate, the state takes the form

$$\overline{B^{\otimes 2}(t=1)} = \frac{1}{d^4-1}\left(d^2\,|SSI\dots I\rangle - |III\dots I\rangle\right), \tag{72}$$

where $I = (1)(2)$ and $S = (12)$ are the identity and swap permutation operators, respectively.

The non-trivial matrix elements in Eq. (67) are

$$T_{SS,IS} = T_{II,IS} = T_{SS,SI} = T_{II,IS} = \frac{d}{d^2+1}.$$

Thus, applying $\mathcal{P}_2$ causes hopping of the domain wall; a single domain wall cannot be destroyed without meeting another domain wall or the boundary, and no new domain walls are created. Thus, the first component of Eq. (72) evolves into a superposition of states with a single domain wall between the S domain and I domain. The hopping occurs in both directions equally according to the dynamical rule Eq. (67).

The desired quantity $\overline{\mathcal{C}^{(2)}}$ is expressed as the overlap of this evolved operator with the final state $d^{-n}\,\langle SS\dots S|\,M^{\otimes 2}$. Aside from the location of the measurement operator M, the final state weights components of $\overline{B^{\otimes 2}(t)}$ by a factor d^{ℓ}, where ℓ is the size of the domain of S. Thus, configurations with large domains of S dominate contributions to $\overline{\mathcal{C}^{(2)}}$.

The combined effect of the dynamics and the final state reweighting can be incorporated by a change of basis to the rescaled operators $\hat{I} = \frac{I}{d}, \hat{S} = \frac{S}{d^2}$. After this change of basis, the dynamical rule Eq. (67) is modified into Eq. (70) with nontrivial matrix elements $\hat{T}$

$$\begin{aligned}\hat{T}_{SS,IS} = \hat{T}_{SS,SI} &= \frac{d^2}{d^2+1},\\ \hat{T}_{II,IS} = \hat{T}_{II,SI} &= \frac{1}{d^2+1}.\end{aligned} \tag{73}$$

This dynamics is Markovian — the total weight is conserved and all matrix elements are positive. The operator coefficients in Eq. (69) after this change of basis can be reinterpreted as probabilities. Unlike the dynamics under T, the dynamics under $\hat{T}$ breaks the symmetry between the two permutations; $\hat{S}$ domains tend to grow and $\hat{I}$ domains tend to shrink.

Let us now specialize to the case where the measurement operator M chosen to be a non-identity Pauli operator on a single site x; we make the x dependence of the correlator explicit by writing the OTOC in this case as $\overline{\mathcal{C}^{(2)}}(x,t)$. In this case, the overlap with the final state can be rewritten as the probability to find the permutation S at site x, so that $\overline{\mathcal{C}^{(2)}}(x,t)$ precisely measures the progress of the growth of the S domain across the system.

In one dimension, for a butterfly operator that starts at the left of the system, the growth of the S domain is described entirely by the location of the right end point of

the domain wall, which undergoes a biased random walk with probability $p = d^2/(d^2+1)$ to the right and $1-p$ to the left. The time dependence of $\overline{\mathcal{C}}$ has the shape of a wavefront that propagates to the right with the average velocity v_B and broadens diffusively with width $\sqrt{2Dt}$. The values of v_B and D for Haar random 1D circuits are directly computable from the random walk hopping probability p:

$$v_B = 2p - 1 = \frac{d^2-1}{d^2+1} \tag{74}$$

$$D = 2p(1-p) = \frac{2d^2}{(d^2+1)^2}. \tag{75}$$

As the probability distribution for the location of the random walker becomes a normal distribution at long times, with mean $x_0 + v_B t$ and standard deviation $\sqrt{2Dt}$, asymptotically $\overline{\mathcal{C}^{(2)}}(x,t)$ becomes an error function. Plotted against a rescaled coordinate

$$u = \frac{x - (x_0 + v_B t)}{\sqrt{2Dt}},$$

the mean OTOC quickly converges to a scaling function that describes the asymptotic behavior, as shown in Fig. S50(a). In higher dimensions, the results are qualitatively similar, however, the width of the front scales with different exponents. It has been argued [29] that these exponents correspond to Kardar-Parisi-Zhang universality class [68].

2. $OTOC^{(1)}$ variance

In contrast to the average, the second moment of $\mathrm{OTOC}^{(1)}$ (first term of Eq. (63)) requires a circuit average of four copies of operator evolution, i.e. in Eq. (65) we set $r = 4$. In this case Eq. (66) projects onto a subspace spanned by $r! = 24$ states labeled by permutations of four elements.

Once again using a non-identity Pauli operator on the left-most site as the initial state $B(t=0)$ of our evolution, after the first layer of Haar random gates the operator is projected into the permutation space

$$\overline{B^{\otimes 4}(t=1)} = \sum_{\alpha \in S_4} c_\alpha \left|\alpha\alpha I \ldots I\right\rangle,$$

creating an initial superposition of domains of different types. As in the case of the mean OTOC, the evolution proceeds by creating a operator front that propagates ballistically with speed v_B and broadens diffusively. Behind the front, where the dynamics is fully scrambled, one always finds a uniform domain of a single permutation or a superposition of uniform domains $\sum_\alpha |\alpha\alpha\alpha \ldots III\rangle$, where the $\ldots$ conceals a potentially complicated superposition of permutations within the front region. Physically, this uniform domain indicates that far behind the front the dynamics is indistinguishable from global scrambling using rth moment quantities.

The desired second moment quantity is computed as the overlap with $d^{-2n} \langle \omega \ldots \omega | M^{\otimes 4}$, where $\omega = (12)(34)$. Ignoring momentarily the effect of the butterfly operator M, the overlap of a uniform domain α of size l with the final state ω is exponential

$$d^{-2n} \langle \omega | \alpha \rangle^l = d^{(2-|\omega^{-1}\alpha|)l}. \tag{76}$$

The effect of the final state competes with the weight of the domain wall which in the leading order corresponds to $(d/(d^2+1))^{t|\alpha|}$ for each moment in time. Domains for which these two effects balance each other out, i.e. that satisfy,

$$2 - |\omega^{-1}\alpha| - |\alpha| = 0, \tag{77}$$

contribute significantly to the variance of OTOC. Only four types of domains satisfy this condition: $\alpha \in \{I, (12), (34), (12)(34)\}$. The contribution of all other domains to the OTOC variance is exponentially suppressed by its boundary. Moreover, the entropy of domain configurations is not sufficient to compensate for this effect: the number of configurations of a domain with perimeter l on a square lattice scales as 2^l, which is insufficient to overcome the weight of its boundary $\left(\frac{d}{d^2+1}\right)^l$, for $d \geq 2$.

Using the rescaled dynamics Eq. (70), the effect of the final state is incorporated throughout the dynamical evolution. In this dynamics, the suppressed domains are unstable and with weight $1 - \mathcal{O}(d^{-2})$ transition into non-suppressed domains, which are stable. As a result, after an initial transient dynamics a typical component of the permutation wavefunction is described as having large domains of only $I, (12), (34)$ and $(12)(34)$. For the choice of a butterfly operator that is a non-identity Pauli operator, the situation is even simpler: a typical configuration has far behind the front $(12)(34)$, a single domain wall transitioning to a domain of (12) or (34) in the front, and a second domain wall to the I domain ahead of the front. The two domain walls each undergo random walks and interact only when colliding.

In the limit $d \to \infty$ the two domain walls may be treated as independent, in which case their joint probability distribution is simply a product. Independent domain

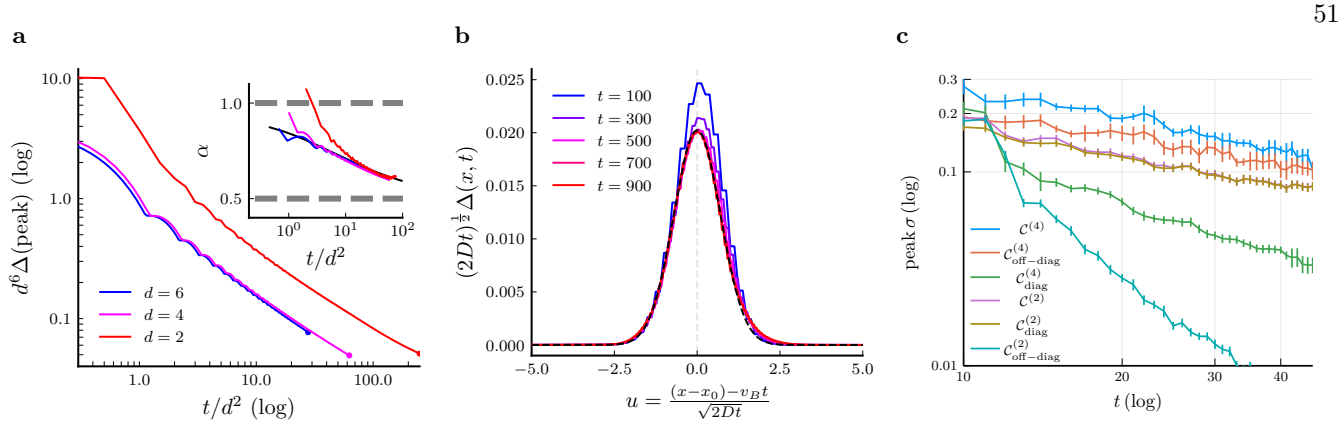


FIG. S49. **Fluctuations of OTOCs for a 1D brickwork circuit with Haar random two-qubit gates.** The butterfly operator B is chosen as a Pauli Z on the left-most site, while the measurement operator is chosen as a Pauli Z on a site x that varies across the 1D chain. The signal propagates from left-to-right with the butterfly velocity v_B in a diffusively growing front. **a**, Peak variance of $\mathcal{C}^{(2)}$ (labeled $\Delta = \sigma^2(\mathcal{C}^{(2)})$) plotted versus time, rescaled for comparison between different values of qudit dimension d. Exponent of the decay α, extracted from the average slope on the log-log plot over a sliding time window, drifts from $1/t$ to $1/\sqrt{t}$ (inset), consistent with the crossover form predicted in Eq. (79). As d is increased, the curves collapse under the rescaling. **b**, Spatial distribution of the variance of $\mathcal{C}^{(2)}$. Asymptotically, when plotted against the rescaled front coordinate u and rescaled by $\sqrt{2Dt}$ the variance collapses onto a curve proportional to a Gaussian with standard deviation $1/\sqrt{2}$ (black dashed curve). **c**. Peak standard deviation of $\mathcal{C}^{(2)}$ and $\mathcal{C}^{(4)}$, as well as the diagonal and off-diagonal components thereof, measured numerically using state vector calculations on up to 36 qubits with a sample of 200 random circuits. The variance of $\mathcal{C}^{(4)}$ undergoes a similar power-law decay as that of $\mathcal{C}^{2)}$. For each circuit, we additionally compute $\mathcal{C}^{(2)}_{\text{diag}}$ and $\mathcal{C}^{(4)}_{\text{diag}}$ by averaging over random Pauli strings inserted at a depth half way through the circuit, as done in the main text, and the corresponding off-diagonal components by subtracting. While $\mathcal{C}^{(2)}_{\text{off-diag}}$ and $\mathcal{C}^{(4)}_{\text{off-diag}}$ both appear to decay as power-laws, $\mathcal{C}^{(2)}_{\text{off-diag}}$ decays more rapidly.

walls are described by the transfer matrix in Eq. (73). The leading order correction corresponds to a collision of the two domain walls, where the following process is possible,

$$(12)(34)I \to \sigma\sigma \to (12)(34)I. \tag{78}$$

Here arrows correspond to applying a layer of two-qudit gates. For independent domain walls this process has the weight $4p^2(1-p)^2$. The leading order correction corresponds to generating intermediate domains $\sigma = \{(14)(23), (13)(24)\}$ which introduces correlations between the domain walls. The respective correction to OTOC variance is of the order of d^{-4}.

Consider two domain walls propagating independently with a single collision point, where the process in Eq. (78) occurs. The probability of moving for time t without changing the relative distance between the pair of domain walls solves $(p^2+(1-p)^2)^t \approx \exp\left(-2t/d^2\right)$. Therefore for very short times $t < d^2$ the collision of domain walls is unlikely. In this case the OTOC variance is dominated by a boundary term at the point of measurement. At longer times, $t > d^2$, domain walls collide in the bulk. The leading order calculation corresponds to a single collision.

These considerations produce the following result for the maximum of the variance Δ over the location of the measure and butterfly qudits as a function of the circuit depth t:

$$\max\Delta \propto \begin{cases} \frac{1}{d^4}\frac{1}{t}, & t < d^2, \\ \frac{1}{d^5}\frac{1}{\sqrt{t}}, & t \geq d^2. \end{cases} \tag{79}$$

This result is valid for $d \gg 1$, but by numerical simulation we show that this result is valid as well for relatively small d. The result of these numerics is detailed in Fig. S49. As both lines of Eq. (79) are consistent with a single scaling form $d^6\Delta = F(t/d^2)$, we confirm the d dependence by plotting $d^6\Delta$ versus the rescaled time t/d^2. This results in a collapse of the data for curves with $d \geq 4$. While $d = 2$ does not collapse onto the same curve as $d \geq 4$, in either case the peak variance is indeed governed at short times by $1/t$ decay and at long times by $1/\sqrt{t}$ decay.

This numerics uses an MPS representation of the permutation wavefunction to exactly compute the dynamics in the permutation subspace, and it uses *time-evolving block decimation* [69] to evolve this wavefunction. Specifically, we use the rescaled dynamics Eq. 70 as the evolution rule, which results in improved convergence for computing the mean and second moment of OTOC. The algorithm uses a truncation parameter $\varepsilon \lesssim 10^{-12}$ which

governs the level of compression in the standard manner, by discarding small singular values s_a whose total $\sum_a s_a^2 \leq \varepsilon$, with the result checked for convergence over several orders of magnitude in ε.

3. *Mean OTOC$^{(2)}$*

As in the case of the variance of OTOC$^{(1)}$, for the analysis of mean OTOC$^{(2)}$ ($\overline{C^{(4)}}$) we use $r = 4$ in Eq. (66). The averaged Haar gates project onto a subspace spanned by $r! = 24$ states labeled by permutations of four elements, and the dynamical rule Eq. (67) is identical. However, the final state changes from $\omega = (12)(34)$ to $\omega = (1234)$; the components of the permutation wavefunction with large overlap on that state undergo an evolution that is qualitatively different. As in the previous sections, the effect of this distinct final state can be incorporated throughout the evolution using the rescaled dynamics Eq. (70), but now using overlaps with the cycle $\omega = (1234)$ for rescaling.

The analysis of stable and unstable domains from the variance of OTOC$^{(1)}$ needs to be modified as follows: the overlap of a domain α with the new final state $d^{-n}\langle\omega\ldots\omega|$ instead results in an exponential weight

$$d^{-l}\langle\omega|\alpha\rangle^l = d^{(3-|\omega^{-1}\alpha|)l}. \tag{80}$$

Domains for which the dynamical weight of the domain wall balances the effect of the final state are ones that satisfy

$$3 - |\omega^{-1}\alpha| - |\alpha| = 0. \tag{81}$$

Many more permutations satisfy this than the 4 in the previous section. They can be characterized easily by noticing that $|\sigma^{-1}\tau|$ is a distance metric on the *Cayley graph*, a graph with permutations as nodes and edges for permutations connected by multiplication of a single transposition. Thus, by the triangle rule, every permutation satisfies

$$|\omega^{-1}\alpha| + |I^{-1}\alpha| \geqslant |I^{-1}\omega| = 3, \tag{82}$$

with the equality satisfied by permutations on a shortest path between $\omega = (1234)$ and the identity I on the Cayley graph. By manual inspection of the Cayley graph, this results in the following list of fourteen stable permutations:

$$I, (12), (13), (14), (23), (24), (34), (123), (124), (134), (234), (12)(34), (14)(23), (1234). \tag{83}$$

In addition, we find that there is a selection rule in the dynamics; after an initial transient time, permutations that have zero overlap with $B(t=0)^{\otimes 4}$ will not appear behind the front. As a result, an initial butterfly operator chosen as a non-identity Pauli evolves into a superposition of only three distinct permutation domains corresponding to the permutations $\alpha = (12)(34)$, $\beta = (14)(23)$, and $\omega = (1234)$, which one can write schematically as

$$\overline{B^{\otimes 4}(t)} \sim \left|\hat{\alpha}\hat{\alpha}\hat{\alpha}\ldots\hat{I}\hat{I}\right\rangle + \left|\hat{\beta}\hat{\beta}\hat{\beta}\ldots\hat{I}\hat{I}\right\rangle - \left|\hat{\omega}\hat{\omega}\hat{\omega}\ldots\hat{I}\hat{I}\right\rangle,$$

using as before the notation $\hat{\sigma}$ to represent a permutation rescaled by its overlap with the final boundary condition $\langle\omega|$. We note that each of these expanding domains becomes a complex superposition of all permutation types — primarily the 14 stable permutations listed above — in the region of the front. The total weight in the collections of terms with a α, β, and ω domain quickly converges exactly to 1, 1, and -1 respectively, which can be derived analytically by considering the average output of a globally Haar random unitary acting on the chosen butterfly operator.

The value of OTOC$^{(2)}$ for the measurement operator M that consists of a non-identity Pauli at site x can be expressed as the total quasi-probability of finding one of 9 permutations at site x — specifically, the 9 permutations τ such that

$$\text{Trace}\left[C^\dagger Z^{\otimes 4}\hat{\tau}\right] = 1.$$

In the 1D chain, we can compute the resulting dynamics with MPS, as in the previous section. In Fig. S50(a), we compare the mean OTOC$^{(1)}$ result with mean OTOC$^{(2)}$. While both transition from 1 to 0 across the operator front, the mean OTOC$^{(2)}$ is sharper and takes a scaling form that deviates from an error function. In Fig. S50(b), we see that the resulting OTOC$^{(2)}$ signal can be split into three parts, corresponding to the $\hat{\alpha}, \hat{\beta}$, and $\hat{\omega}$ domains behind the front. Each of these fronts grows with the same butterfly velocity v_B that is relevant for $k = 2$. In Fig. S50(c), we show that the front is a complex superposition of permutations requiring a large bond dimension to represent accurately with an MPS. For truncation error $\varepsilon = 10^{-12}$, the maximum bond dimension saturates around 1750; computations with smaller bond dimensions show significant error buildup in the mean OTOC$^{(2)}$ values over time. For two dimensional systems, MPS requires a cost exponential in width to capture the front, which results in unfeasibly large bond dimensions

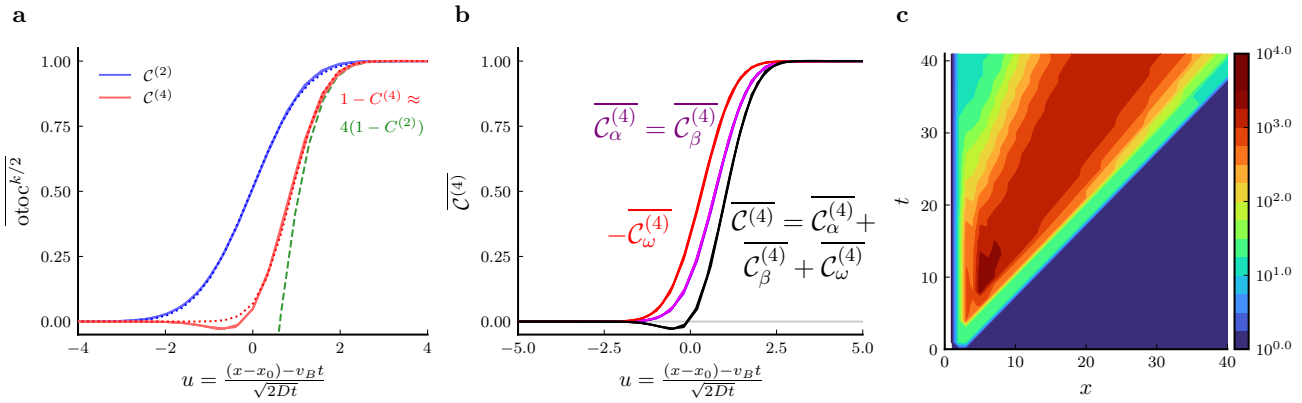


FIG. S50. **a**, Comparing the mean value of $\text{OTOC}^{(k/2)}$ for $k = 2$ and $k = 4$ for a 1D brickwork circuit with Haar random two-qubit gates. The butterfly operator O is chosen as a Pauli Z on the left-most site, while the measurement operator is chosen as a Pauli Z on a site x that varies across the 1D chain. The signal propagates from left-to-right with the butterfly velocity v_B in a diffusively broadening front; plotted against the rescaled coordinate u, the signal collapses into a universal shape. The scaling limit of $k = 2$ is well approximated by an error function (dashed lines), while $k = 4$ deviates from an error function. In the leading edge of the front, the mean $\text{OTOC}^{(k/2)}$ satisfies $1 - \mathcal{C}^{(k)} \approx (\frac{k}{2})^2(1 - \mathcal{C}^{(2)})$. This property is a manifestation of the correlation operator spectrum when it consists of small arcs, as shown in Fig. S55. In the middle of the front, $\mathcal{C}^{(4)}$ oscillates, qualitatively similarly to what is seen in the random matrix model Fig. S54. **b**, In the permutation representation, a superposition of three growing domains is created, with domains of $\alpha = (12)(34)$, $\beta = (14)(23)$, and $\omega = (1234)$ which in the scaling limit comprise the entire signal. The signal is separated into pieces $\mathcal{C}_\alpha^{(4)}$ by projecting the left-most site into the permutation α, with the total well approximated by $\mathcal{C}_\alpha^{(4)} + \mathcal{C}_\beta^{(4)} + \mathcal{C}_\omega^{(4)}$, the later of which is negative. **c**, Shown here is the MPS bond dimensions in the mean $\text{OTOC}^{(2)}$ calculation. The peak bond dimension saturates in time and follows the position of the front, allowing for an efficient simulation of the mean $\text{OTOC}^{(2)}$ in 1D. However, the front is a complex superposition of permutation configurations, requiring an bond dimension of over 1500 to represent accurately. A two dimensional extended mean $\text{OTOC}^{(2)}$ front (not shown) is far more complex and beyond the capabilities of our MPS simulations to capture.

for computing mean $\text{OTOC}^{(2)}$ even in narrow strip geometries.

B. Criticality in the correlation operator spectrum

1. *Overview: spectrum and moments of the correlation operator*

The goal of this supplement is to elucidate spectral properties of the "correlation operator" $C(t) = MB(t)$. We will show that the spectrum of $C(t)$ undergoes a dynamical phase transition as a function of time t. On one side of the transition, the spectrum is gapped, while on the other side it is gapless. We will demonstrate how the phase transition is reflected in out-of-time-ordered correlators (OTOCs).

Explicitly, the correlation operator between local observables M and B is given by

$$C(t) = MU^\dagger(t)BU(t). \tag{84}$$

Here $U(t)$ is a unitary matrix, e.g., generated by an application of a random circuit of depth t (comprising single-qubit and local two-qubit operations) to n qubits. Hereafter, we will focus on the case in which M and B are Pauli matrices acting on two different qubits, $M = Z_M$ and $B = Z_B$. This choice makes the correlation operator unitary, $C^\dagger C = \mathbb{1}$. It also leads to the "parity" symmetry $C^\dagger = Z_M C Z_M$.

To explain what makes the spectral structure of $C(t)$ interesting, we first note that, due to the unitarity, all eigenvalues λ of $C(t)$ lie on a unit circle in the complex plane, $\lambda = e^{i\varphi}$. The distribution of eigenvalues evolves with t. It is in this evolution that the phase transition occurs, see Fig. S51. Hints towards the existence of the phase transition can be seen from the comparison of the two extreme cases. At $t = 0$, unitary $U = \mathbb{1}$. This turns $C(0)$ into a product of commuting operators Z_M and Z_B, which can be straightforwardly diagonalized. There are only two distinct eigenvalues: $\lambda = \pm 1$. Each of the two has a macroscopic degeneracy $D_\text{H}/2$ determined by the Hilbert space dimension $D_\text{H} = 2^n$.

In the opposite limit, $t \to \infty$, $U(t)$ becomes a fully scrambling unitary. This randomizes $C(t)$, and the eigenvalues of the latter spread uniformly across the circle.

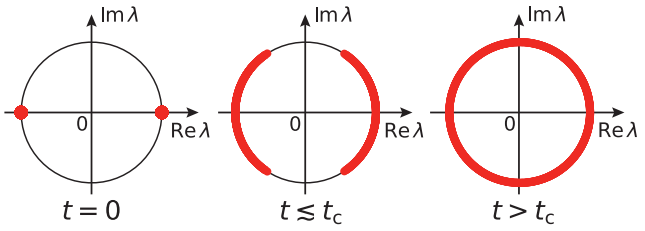


FIG. S51. **Sketch of the evolution of the correlation operator spectrum with t across the phase transition point $t_\star$.** (See Figs. S52 and S55 for fine details). At $t = 0$, there are only two distinct eigenvalues, $\lambda = \pm 1$; each has a macroscopic degeneracy $D_{\rm H}/2$ determined by the Hilbert space dimension $D_{\rm H}$. At finite $t < t_\star$, the spectrum broadens into two arcs. However, it remains gapped. The gap vanishes at $t \geq t_\star$.

The mean angular spacing between the levels can be estimated as $\delta\varphi \sim 1/D_{\rm H} = 2^{-n}$; it is exponentially small in the system size. Thus, for all practical purposes the spectrum can be viewed as a gapless one.

On the other hand, the spectrum cannot immediately turn gapless with the increase of t from zero. Indeed, at small but finite t, perturbation theory is justified. It leads to the splitting of the degeneracy but only by a finite value. The spectrum, therefore, consists of two non-overlapping arcs, see Fig. S51. The size of the arcs increases with t until, eventually, the gap closing occurs. This defines a critical point $t_\star$ of the spectral phase transition.

To elucidate the evolution of the spectrum across $t_\star$, we introduce the spectral density

$$S(\varphi) = \frac{1}{D_{\rm H}} \sum_{i=1}^{D_{\rm H}} \delta(\varphi - \varphi_i) \tag{85}$$

[in passing, we note that $S(\varphi) = S(-\varphi)$ due to the parity symmetry]. Below, we will develop a theory for the spectral density in an analytically-solvable random matrix theory model, see Section IV B 2; we will also present numerical results on quantum circuits composed of random two-qubit gates, see Section IV B 5.

The reconstruction of the spectrum occurring across $t = t_\star$ is manifested in the dependence of higher-order OTOCs, $\mathcal{C}^{(2k)}(t)$, on k. To see this, note that the higher-order OTOCs are nothing but the Fourier harmonics of the spectral density $S(\varphi)$

$$\mathcal{C}^{(2k)}(t) = \tfrac{1}{D_{\rm H}} \mathrm{Trace}[C^{2k}(t)] = \int_{-\pi}^{\pi} d\varphi\, S(\varphi) \cos(2k\varphi). \tag{86}$$

The asymptotic behavior of the harmonics at $k \gg 1$ reveals the presence of discontinuities in the function $S(\varphi)$; it is thus different below and above $t_\star$. In the gapped phase, $t < t_\star$, $S(\varphi)$ is supported only within the spectral arcs [see Fig. S51], and its derivatives are discontinuous at the arcs edges. This translates into a slow power-law decay of the harmonics with k,

$$\mathcal{C}^{(2k)}(t) \propto \frac{\cos(\omega k - \xi)}{k^\alpha}, \qquad t < t_\star. \tag{87}$$

The decay is accompanied by oscillations [ω and ξ depend smoothly on $t_\star - t$]. A power-law dependence persists to the critical point,

$$\mathcal{C}^{(2k)}(t) \propto \frac{(-1)^k}{k^{\alpha_{\rm c}}}, \qquad t = t_\star, \tag{88}$$

although the exponent changes to $\alpha_{\rm c} \neq \alpha$. The behavior is different in the gapless phase, $t > t_\star$, where $S(\varphi)$ becomes a smooth function. The smoothness results in an exponential dependence

$$\mathcal{C}^{(2k)}(t) \propto (-1)^k \exp(-\nu k), \qquad t > t_\star, \tag{89}$$

with $\nu \equiv \nu(t - t_\star) > 0$. The change in the asymptotic behavior across $t_\star$ is a direct signature of the phase transition that can be observed by measuring OTOCs of different order.

The remainder of the supplement is organized in the following way. We describe the random-matrix theory model of the spectral phase transition in Section IV B 2. In Section IV B 3, we find OTOCs in this model. In particular, we compare the asymptotic behavior of the high-order OTOCs below, at, and above the critical point, see Section IV B 3 a. Section IV B 4 contains detailed derivations of the central equations of the random-matrix theory. Lastly, in Section IV B 5, we demonstrate the existence of the phase transition in a realistic random quantum circuit.

2. *Random matrix model of the spectral phase transition*

In this section, we present a solvable random-matrix based model allowing one to capture the evolution of spectrum of the correlation operator $C = Z_M U^\dagger Z_B U$ with the increase of randomness in U. We will use this model to elucidate the critical behavior in the spectral density $S(\varphi)$, and show how the criticality is reflected in the higher-order OTOCs.

Specifically, we consider a model in which the evolution operator U is a random $D_{\rm H} \times D_{\rm H}$ unitary matrix. To fully define the model, we need to describe an ensemble from which U is sampled. A natural choice would be to first generate a random Hamiltonian H from a Gaussian unitary ensemble (GUE),

$$\langle H_{ij} \rangle = 0, \tag{90a}$$

$$\langle H_{ij} H_{kl} \rangle = \frac{\gamma}{D_{\rm H}} \delta_{il} \delta_{jk}, \quad i,j,k,l = 1, \ldots, D_{\rm H}, \tag{90b}$$

and then introduce $U = \exp(-2iH)$ as the respective evolution operator [we normalized the correlator of H by the Hilbert space dimension to make the thermodynamic limit $D_{\rm H} \to \infty$ well-defined]. There is a correspondence between the model parameter γ and the evolution time t of Eq. (84) and Fig. S51; we will clarify this correspondence in Section IV B 5. In principle, the described choice of U would allow one to capture the crossover between $U = \mathbb{1}$ and a fully scrambled U by varying γ. The problem with this approach is the exponential dependence of U on H; it makes finding the spectral density $S(\varphi)$ of C analytically a difficult task. Instead, we will consider an ensemble of unitaries obtained as a Cayley transform of GUE:

$$U = \frac{\mathbb{1} - iH}{\mathbb{1} + iH}. \tag{91}$$

For small γ, U defined in this way is close to $U = \exp(-2iH)$. Although the respective ensembles become significantly different for $\gamma \gtrsim 1$, we will see that the phase transition occurs already at $\gamma = 1/3$. The main advantage of the ensemble in Eq. (91) is that for such an ensemble it is possible to find the spectral density $S(\varphi)$ analytically [70].

To access $S(\varphi)$, we introduce the Green's function of the correlation operator:

$$G(\lambda) = \frac{1}{\lambda \mathbb{1} - C}. \tag{92}$$

The Green's function contains the full spectral information about C; in particular, the spectral density can be represented as

$$S(\varphi) = \frac{1}{2\pi D_{\rm H}} \sum_{s=\pm 1} s\lambda \left\langle \mathrm{Trace}[G(\lambda(1 + s\,0^+))] \right\rangle \big|_{\lambda = e^{i\varphi}}, \tag{93}$$

where 0^+ is an infinitesimal positive number and $\langle \ldots \rangle$ denotes the averaging over the realizations of H.

The introduction of ensemble averaging deserves a comment. On the one hand, it is this ensemble averaging that makes the problem tractable. The averaging washes out fine features of the level distribution, and leads to a well-behaved, smooth density function $S(\varphi)$. On the other hand, it raises a question of the importance of sample-to-sample fluctuations. In fact, for our random-matrix theory model, sample-to-sample fluctuations are negligible; this is in full similarity to standard random-matrix theory behavior where the fluctuations are small in $1/D_{\rm H}$.

How does one carry out an ensemble averaging in Eq. (93)? Were C linear in H, one could expand the right-hand side of Eq. (92) in powers of C, and straightforwardly perform the averaging term by term using Eq. (90). However, C depends on H in a non-linear way; this makes a direct realization of such a program laborious.

A key property of ensemble (91) is that for this ensemble the task of finding $\langle \mathrm{Trace}[G(\lambda)] \rangle$ can actually be mapped onto inverting and averaging an auxiliary matrix linear in H. The mapping procedure closely follows Ref. 70, and we explain it in detail in a later section [Section IV B 4]. For now, we only present a final result for $\langle \mathrm{Trace}[G(\lambda)] \rangle$ valid in the limit of $D_{\rm H} \gg 1$. Using a variant of a self-consistent Born approximation tailored for the problem at hand, we will show that

$$\langle \mathrm{Trace}[G(\lambda)] \rangle = \frac{D_{\rm H}}{2} \sum_{r=\pm 1} \frac{1}{\lambda - \left(\frac{\varepsilon + i}{\varepsilon - i}\right)^2 r}. \tag{94}$$

Here $\varepsilon \equiv \varepsilon(\lambda)$ is the "self-energy" function. The self-energy satisfies the self-consistency relation; in a parameterization $\lambda = e^{i\varphi}$ natural for evaluating $S(\varphi)$ [cf. Eq. (93)], this relation reads

$$\varepsilon(\varphi) = \frac{\gamma}{2} \Big(\frac{\varepsilon(\varphi) - \cot(\varphi/2)}{1 - \varepsilon^2(\varphi) + 2\varepsilon(\varphi)\cot(\varphi/2)} + \frac{\varepsilon(\varphi) + \tan(\varphi/2)}{1 - \varepsilon^2(\varphi) - 2\varepsilon(\varphi)\tan(\varphi/2)} \Big). \tag{95}$$

One immediate consequence of this equation is that the average spectral density satisfies $S(\varphi + \pi) = S(\varphi)$. This feature results from Eq. (90a). The other consequence of Eq. (95) is that $S(\varphi) = S(-\varphi)$, which is nothing but the exact parity symmetry discussed after Eq. (84).

Let us now use Eqs. (93)–(95) to describe the evolution of the spectral density $S(\varphi)$ with γ, and elucidate the critical behavior in it.

Weak randomness, $\gamma \ll 1$ — To start off, it is useful to find $S(\varphi)$ in the limit of weak randomness, $\gamma \ll 1$. Without randomness, $D_{\rm H}/2$ eigenvalues of C are equal to $+1$, while the remaining $D_{\rm H}/2$ are equal to -1. The

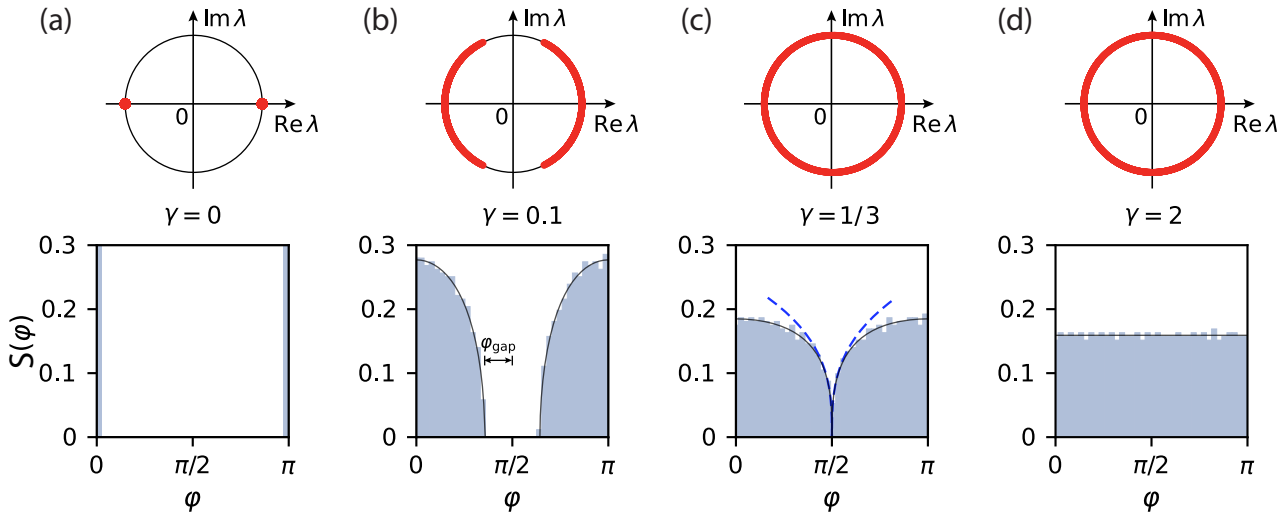


FIG. S52. **Evolution of the correlation operator spectrum in the random-matrix theory model.** **a**, At $\gamma = 0$, there are only two distinct eigenvalues $\lambda = \pm 1$. Each has a macroscopic degeneracy of $D_{\rm H}/2$, where $D_{\rm H}$ is the Hilbert space dimension. **b**, At a finite γ, the spectrum broadens into two arc (histogram is produced by numerically diagonalizing the random matrix C for $D_{\rm H} = 2048$; solid black curve is obtained via a numeric solution of Eqs. (94) and (95)]. **c**, The arcs merge at the critical point $\gamma = \gamma_\star = 1/3$. Dashed blue curve depicts the asymptotic result for $S(\varphi)$ near $\varphi = \pi/2$ [see Eq. (107)). **d**, Spectral density becomes uniform at $\gamma = 2$. As we will see, the evolution of the correlation operator spectrum in realistic quantum circuits with local, random gates is qualitatively similar to the one presented here (cf. Fig. S55).

introduction of randomness lifts this macroscopic degeneracy; however, as long as the randomness is weak, the eigenvalues remain concentrated near $\lambda = \pm 1$. Keeping the symmetry of $S(\varphi)$ in mind, we focus on the modification of the eigenvalue density $S(\varphi)$ near $\varphi = 0$.

For $\gamma \ll 1$ and $\varphi \ll 1$, the self-energy function is itself small, $\varepsilon(\varphi) \ll 1$ (as will become apparent momentarily). This allows one to simplify Eq. (95), and bring it to a compact form

$$\varepsilon(\varphi) = -\frac{\gamma}{2}\frac{1}{2\varepsilon(\varphi) + \varphi/2}. \tag{96}$$

A solution of the equation for $\varepsilon(\varphi)$ is straightforward:

$$\varepsilon(\varphi) = -\frac{\varphi}{8} \pm \frac{i}{8}\sqrt{16\gamma - \varphi^2}. \tag{97}$$

Combining this result with Eqs. (93) and (94), we find a Wigner semicircle distribution of levels:

$$S(\varphi \ll 1) = \frac{1}{16\pi\gamma}\sqrt{16\gamma - \varphi^2}\cdot\Theta(16\gamma - \varphi^2) \tag{98}$$

[here $\Theta(x)$ is a step function]. In addition to a semicircle centered at $\varphi = 0$, $S(\varphi)$ has a semicircle around $\varphi = \pi$, of the same width.

The above shows that the eigenvalues of C form two disjoint arcs on a unit circle, see Fig. S52(b). Outside of the arcs, the density of levels is negligible; the spectrum has a gap. The arcs expand with the increase of γ [see Eq. (98)]. The expansion culminates at a critical value $\gamma_\star$, at which the arcs merge and the spectrum becomes gapless [Fig. S52(c)].

Due to the symmetry of $S(\varphi)$, the closing of the gap occurs at $\varphi = \pm\pi/2$. One can find the critical point $\gamma_\star$ by examining the structure of solutions of Eq. (95) for $\varepsilon(\pm\pi/2)$. The gap closing corresponds to the appearance of complex (as opposed to purely real) solutions for this quantity. From this condition, we find $\gamma_\star = 1/3$. With that, we proceed to the critical behavior of $S(\varphi)$.

Critical behavior of the spectral density — At $\gamma = \gamma_\star = 1/3$ and $\varphi = \pi/2$, the self-energy function vanishes, $\varepsilon = 0$. To study the critical behavior of the spectral density $S(\varphi)$, we find how ε deviates from zero with $\gamma - \gamma_\star$ and $\varphi - \pi/2$. The structure of these deviations directly translates into a respective structure of $S(\varphi)$. Indeed, it follows from Eqs. (93) and (94) that the spectral density satisfies

$$S(\varphi) = -\frac{2}{\pi}\mathrm{Im}\,\varepsilon(\varphi), \tag{99}$$

as long as the self-energy is small, $|\varepsilon| \ll 1$.

Performing an expansion in the deviations $\gamma - \gamma_\star \ll 1$

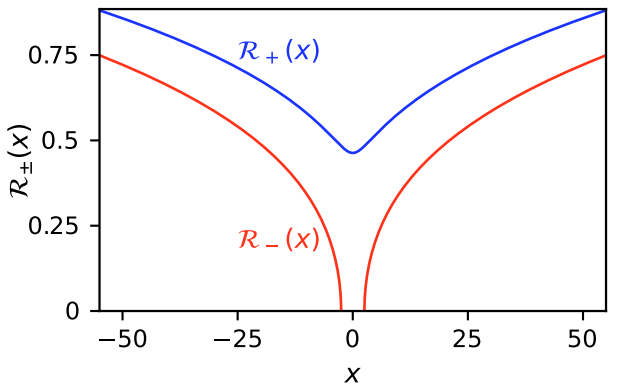


FIG. S53. **Scaling functions $\mathcal{R}_{\pm}(x)$ characterizing the behavior of the spectral density in the vicinity of the critical point**. See Eqs. (102)–(104).

and $\varphi - \pi/2 \ll 1$ in Eq. (95), we obtain the following equation for $\varepsilon(\varphi)$:

$$(\varphi - \pi/2) + 9\varepsilon(\varphi)(\gamma - \gamma_\star) + 17\varepsilon^3(\varphi) = 0. \tag{100}$$

This is a cubic equation, and it can be solved explicitly. At this point though, it is more instructive to examine its general scaling properties.

To implement the scaling analysis, we note that Eq. (100) can be made "dimensionless" by introducing rescaled variables $\tilde{\varepsilon} \equiv |\gamma - \gamma_\star|^{1/2}\varepsilon$ and $\tilde{\varphi} \equiv |\gamma - \gamma_\star|^{3/2}(\varphi - \pi/2)$. In terms of these variables, the equation acquires the form

$$\tilde{\varphi} \pm 9\tilde{\varepsilon} + 17\tilde{\varepsilon}^3 = 0, \tag{101}$$

where the $+$ and $-$ signs correspond to $\gamma > \gamma_\star$ and $\gamma < \gamma_\star$, respectively. Clearly, the solution for $\tilde{\varepsilon}$ depends only on a rescaled angle $\tilde{\varphi}$. This observation stipulates a scaling relation for the self-energy: $\varepsilon(\varphi) = (\gamma - \gamma_\star)^{1/2}\mathcal{E}\big([\varphi - \pi/2]/(\gamma - \gamma_\star)^{3/2}\big)$. Thus, $S(\varphi)$ obeys a scaling law too [cf. Eq. (99)]:

$$S(\varphi) = \begin{cases} (\gamma - \gamma_\star)^{1/2}\mathcal{R}_+\big(\frac{\varphi - \pi/2}{(\gamma - \gamma_\star)^{3/2}}\big), & \gamma > \gamma_\star, \\ (\gamma_\star - \gamma)^{1/2}\mathcal{R}_-\big(\frac{\varphi - \pi/2}{(\gamma_\star - \gamma)^{3/2}}\big), & \gamma < \gamma_\star. \end{cases} \tag{102}$$

The scaling functions are different above and below the phase transition point; they follow from the solution of the cubic equation, Eq. (101). Above the transition, $\gamma > \gamma_\star$, we obtain

$$\mathcal{R}_+(x) = \frac{\sqrt{3}}{34^{1/3}\pi}\Big[\Big(\sqrt{x^2 + \tfrac{108}{17}} + x\Big)^{1/3} + \Big(\sqrt{x^2 + \tfrac{108}{17}} - x\Big)^{1/3}\Big]. \tag{103}$$

Notice that $\mathcal{R}_+(x) > 0$ for all x; this feature reflects that the spectrum is gapless at $\gamma > \gamma_\star$.

The structure of the scaling function changes below the transition point due to the gap opening. The magnitude of the gap can be found by a direct inspection of Eq. (101). A simple analysis of this equation shows that there exists a parameter region $\tilde{\varphi} \in [-\sqrt{108/17}, \sqrt{108/17}]$ in which all three solutions for $\tilde{\varepsilon}$ are real. In this region, the spectral density vanishes [cf. Eq. (99)]. The scaling function describing the gapped spectrum at $\gamma < \gamma_\star$ reads

$$\mathcal{R}_-(x) = \frac{\sqrt{3}}{34^{1/3}\pi}\Big[\Big(|x| + \sqrt{x^2 - \tfrac{108}{17}}\Big)^{1/3} - \Big(|x| - \sqrt{x^2 - \tfrac{108}{17}}\Big)^{1/3}\Big] \cdot \Theta\Big(|x| - \sqrt{\tfrac{108}{17}}\Big). \tag{104}$$

Eqs. (102)–(104) give a complete description of the behavior of spectral density $S(\varphi)$ in the vicinity of the critical point. The behavior of scaling functions $\mathcal{R}_{\pm}(x)$ is demonstrated in Fig. S53.

Let us use the derived scaling functions to provide a set of explicit results for the dependence of the spectrum of C on $\gamma - \gamma_\star$ and φ. First, by combining Eqs. (102) and (104), we obtain the dependence of the spectral gap on $\gamma_\star - \gamma$:

$$\varphi_{\rm gap} = \sqrt{\tfrac{108}{17}}(\gamma_\star - \gamma)^{3/2}. \tag{105}$$

Near the edge of the gap, $0 < |\varphi - \pi/2| - \varphi_{\rm gap} \ll \varphi_{\rm gap}$, the spectral density has a square root behavior,

$$S(\varphi) = \frac{2}{\pi}\left(\frac{|\varphi - \frac{\pi}{2}| - \varphi_{\rm gap}}{3\sqrt{51(\gamma_\star - \gamma)}}\right)^{1/2} \cdot \Theta\big(|\varphi - \tfrac{\pi}{2}| - \varphi_{\rm gap}\big). \tag{106}$$

Eq. (105) and Eq. (106) elucidate how the arcs in the spectrum of C approach each other as $\gamma \to \gamma_\star$.

At the critical point, $\gamma = \gamma_\star$, the gap is absent but $S(\pi/2) = 0$. It is natural to ask how $S(\varphi)$ deviates from zero when φ is tuned away from $\pi/2$. By assessing the asymptotic behavior of scaling functions $\mathcal{R}_{\pm}(x)$ at $|x| \gg 1$, we find

$$S(\varphi) = \frac{\sqrt{3}}{17^{1/3}\pi}|\varphi - \pi/2|^{1/3}. \tag{107}$$

In Fig. S52(c), we compare this prediction to an explicit diagonalization of matrix C, as well as to a direct numerical solution of Eqs. (93)–(95). The three results are in a perfect agreement.

Lastly, we describe how $S(\varphi=\pi/2)$ departs from zero with the increase of γ above $\gamma_\star$. Setting $\varphi=\pi/2$ in Eq. (102), we arrive to

$$S(\pi/2)=\frac{1}{\pi}\sqrt{\tfrac{36}{17}}(\gamma-\gamma_\star)^{1/2}\cdot\Theta(\gamma-\gamma_\star), \tag{108}$$

where a specific numeric coefficient follows from Eq. (103).

Eqs. (105)–(108) define a set of critical exponents: 3/2 for the dependence of the gap on $\gamma_\star-\gamma$; 1/2 for the behavior of $S(\varphi)$ near the gap edge; 1/3 for the φ-dependence of $S(\varphi)$ at the critical point; and 1/2 for the build up of the spectral density at $\varphi=\pm\pi/2$ with $\gamma-\gamma_\star$.

Spectral density at $\gamma=2$ The spectral density of C becomes more and more uniform with the increase of γ above its critical value $\gamma_\star=1/3$. This process completes at $\gamma=2$, where $S(\varphi)$ becomes constant. To see this, we note that at $\gamma=2$ Eq. (95) admits a φ-independent solution $\varepsilon(\varphi)=\pm i$. Its use in Eqs. (93) and (94) leads to $S(\varphi)=1/(2\pi)$. A uniform $S(\varphi)$ can be rationalized by noting that, at $\gamma=2$, the ensemble defined by Eq. (91) mimics the scrambling property $\langle U\rangle=0$ of the circular unitary ensemble (i.e., the Haar-random distribution).

3. *OTOCs in the random-matrix model*

We succeeded in showing the existence of a phase transition that occurs in the spectrum $S(\varphi)$ of the correlation operator C with a variation of γ. It is important to realize, though, that $S(\varphi)$ itself cannot be accessed directly on a quantum computer; one can only measure its moments, i.e., OTOCs [see Eq. (86)]. In this section, we find OTOCs in the random-matrix theory model of Section IV B 2. We will obtain explicit expressions describing the evolution of the low-order OTOCs with γ; we will also elucidate signatures of the spectral phase transition in the dependence of $\mathrm{OTOC}^{(k)}\equiv\mathcal{C}^{(2k)}$ on k. Throughout this section, we focus on ensemble-averaged quantities; this is justified by the fact that the spectral density $S(\varphi)$ is self-averaging in our model.

Derivation of the γ-dependence of the low-order OTOCs — As follows from Eq. (92), individual OTOCs can be extracted from the Laurent expansion of the Green's function:

$$\tfrac{1}{D_\mathrm{H}}\langle\mathrm{Trace}[G(\lambda)]\rangle=\sum_{k=0}^{\infty}\frac{1}{\lambda^{2k+1}}\mathcal{C}^{(2k)}. \tag{109}$$

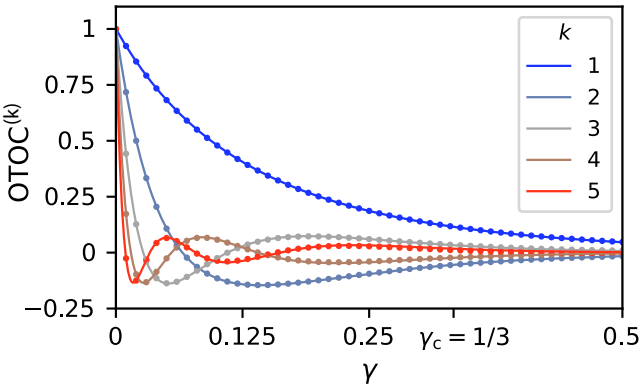


FIG. S54. **Dependence of $\mathcal{C}^{(2k)}$ on γ for $k=1,\ldots,5$.** Solid curves are produced using analytical formulas (Eqs. (111)–(113), and their generalizations for $k=4$ and 5). Dots depict the results obtained by a direct diagonalization of a matrix C for $D_\mathrm{H}=1024$.

[the terms with odd powers of λ cancel due to the symmetry of the spectrum $S(\varphi)=S(\varphi+\pi)$]. We carry out the Laurent expansion with the help of Eqs. (94) and (95). First, we iteratively solve Eq. (95) for the self-energy at $\lambda\to\infty$; this allows us to represent it as a series $\varepsilon(\lambda)=\varepsilon_0+\varepsilon_2/\lambda^2+\varepsilon_4/\lambda^4+\ldots$. For example, ε_0 is given by

$$\varepsilon_0=-\frac{i}{2}\Big(\sqrt{1+4\gamma}-1\Big). \tag{110}$$

The expression for ε_2 and higher-order coefficients are cumbersome, and we refrain from presenting them here. Then, we obtain the OTOCs by substituting the found $\varepsilon(\lambda)$ into Eq. (94), and expanding the latter in $1/\lambda$. With due diligence, this procedure can be used to find $\mathcal{C}^{(2k)}$ with any given k. For $k=1,2$ and 3, we find

$$\mathcal{C}^{(2)}(\gamma)=\left(\frac{3-\sqrt{4\gamma+1}}{1+\sqrt{4\gamma+1}}\right)^4, \tag{111}$$

$$\mathcal{C}^{(4)}(\gamma)=\left(\frac{3-\sqrt{4\gamma+1}}{1+\sqrt{4\gamma+1}}\right)^6 \frac{(4\gamma-22)\sqrt{1+4\gamma}+26-24\gamma}{\sqrt{1+4\gamma}(1+\sqrt{1+4\gamma})^2}, \tag{112}$$

$$\mathcal{C}^{(6)}(\gamma)=8\left(\frac{3-\sqrt{4\gamma+1}}{1+\sqrt{4\gamma+1}}\right)^8 \frac{(241+240\gamma-78\gamma^2+8\gamma^3)\sqrt{1+4\gamma}-239-814\gamma+736\gamma^2-96\gamma^3}{(1+4\gamma)^{3/2}(1+\sqrt{1+4\gamma})^4}. \tag{113}$$

As expected, $\mathcal{C}^{(2)}=\mathcal{C}^{(4)}=\mathcal{C}^{(6)}=1$ at $\gamma=0$. In the regime where U is fully scrambling, $\gamma=2$, we obtain $\mathcal{C}^{(2k)}=0$; this is in agreement with a uniform distribution $S(\varphi)$ found above. Analytical expressions for higher-order OTOCs ($k=4,5,\dots$) quickly become unwieldy, so we do not present formulas with $k\geq 4$ explicitly.

Fig. S54 shows the dependence of $\mathcal{C}^{(2k)}$ with $k=1,\dots,5$ on γ. In contrast to the lowest-order $\mathcal{C}^{(2)}$, OTOCs with $k\geq 2$ are non-monotonic, sign-alternating functions of γ.

a. Signature of the phase transition in the OTOCs. Fig. S54 may create an impression that the phase transition occurring at $\gamma_\star=1/3$ is undetectable on the level of OTOCs. This is not correct. As discussed in Section IV B 1, the phase transition can be revealed by assessing the asymptotic dependence of OTOCs on k. Here we find the exponents α and α_c characterizing this dependence in our random-matrix theory model [cf. Eqs. (87) and (88)]. We also find signatures of the criticality in the dependence of ω and ν [see Eqs. (87) and (89)] on $\gamma-\gamma_\star$.

To set the stage, it is useful evaluate $\mathcal{C}^{(2k)}$ deep in the gapped phase, $\gamma\ll\gamma_\star$. In this case, the distribution of levels is a Wigner semicircle [see Eq. (98)]. For such a simple distribution, all of the OTOCs can be found analytically. Using Eq. (98) in Eq. (86) and computing the resulting integral, we obtain

$$\mathcal{C}^{(2k)}(\gamma)=\frac{J_1(8k\sqrt{\gamma})}{4k\sqrt{\gamma}}, \tag{114}$$

where $J_1(x)$ is the Bessel function. At large k, this expression turns into

$$\mathcal{C}^{(2k)}(\gamma)\approx\frac{\cos\left(8\sqrt{\gamma}k-3\pi/4\right)}{8\sqrt{\pi}\gamma^{3/4}k^{3/2}}. \tag{115}$$

The dependence of OTOC on k is oscillatory; the frequency of oscillations $\omega=8\sqrt{\gamma}$ depends on γ. Eq. (115) fixes $\alpha=3/2$ for an exponent describing the power-law decay of the higher-order OTOCs in the gapped phase [cf. Eq. (87)].

We note that the value $\alpha=3/2$ persists beyond the applicability of Eq. (115); in fact, it remains the same throughout the gapped phase. This value reflects the character of the edge singularity in the spectral density at $\gamma<\gamma_\star$, $S(\varphi)\propto(|\varphi-\pi/2|-\varphi_{\rm gap})^{1/2}\cdot\Theta(|\varphi-\pi/2|-\varphi_{\rm gap})$.

The character of the edge singularity changes at the critical point, $\gamma=\gamma_\star$. There, $S(\varphi)\propto|\varphi-\pi/2|^{1/3}$ [see Eq. (107)]. The exponent governing the dependence of OTOCs on k jumps to $\alpha_c=4/3$.

In the gapless phase, $\gamma>\gamma_\star$, $S(\varphi)$ becomes a smooth function. This results in the exponential decay of its harmonics with k, $\mathcal{C}^{(2k)}\propto\exp(-\nu k)$. Near the critical point, the dependence of the coefficient ν on $\gamma-\gamma_\star$ follows from scaling law (102). Using this law together with Eq. (86), we obtain $\nu\propto(\gamma-\gamma_\star)^{3/2}$. A similar argument applied the gapped side of the transition gives $\omega-\pi\propto(\gamma_\star-\gamma)^{3/2}$.

4. *Derivation of Eqs. (94) and (95)*

Let us now explain how we derived Eqs. (94) and (95) that were central for the analysis of spectral phase transition.

The object needed to compute $S(\varphi)$ is the trace of the ensemble-averaged Green's function:

$$\langle\mathrm{Trace}_{\rm H}[G(\lambda)]\rangle=\left\langle\mathrm{Trace}_{\rm H}\left[\frac{1}{\lambda-Z_MU^\dagger Z_BU}\right]\right\rangle \tag{116}$$

[we explicitly indicated that the trace is evaluated over the $D_{\rm H}$-dimensional Hilbert space; this is done to distinguish $\mathrm{Trace}_{\rm H}$ from the traces over the auxiliary spaces which we introduce below]. At face value, the ensemble averaging in Eq. (116) is hard to carry out. The reason is the non-linear dependence of $Z_MU^\dagger Z_BU$ on H [cf. Eq. (91)]. A method for simplifying problems like this one was recently developed in Ref. 70. It was shown that evaluation of $\langle\mathrm{Trace}_{\rm H}[G(\lambda)]\rangle$ for ensemble (91) can be reduced to a simpler problem involving inversion and averaging a matrix *linear* in H. The latter problem is amenable to standard diagrammatic technique [71].

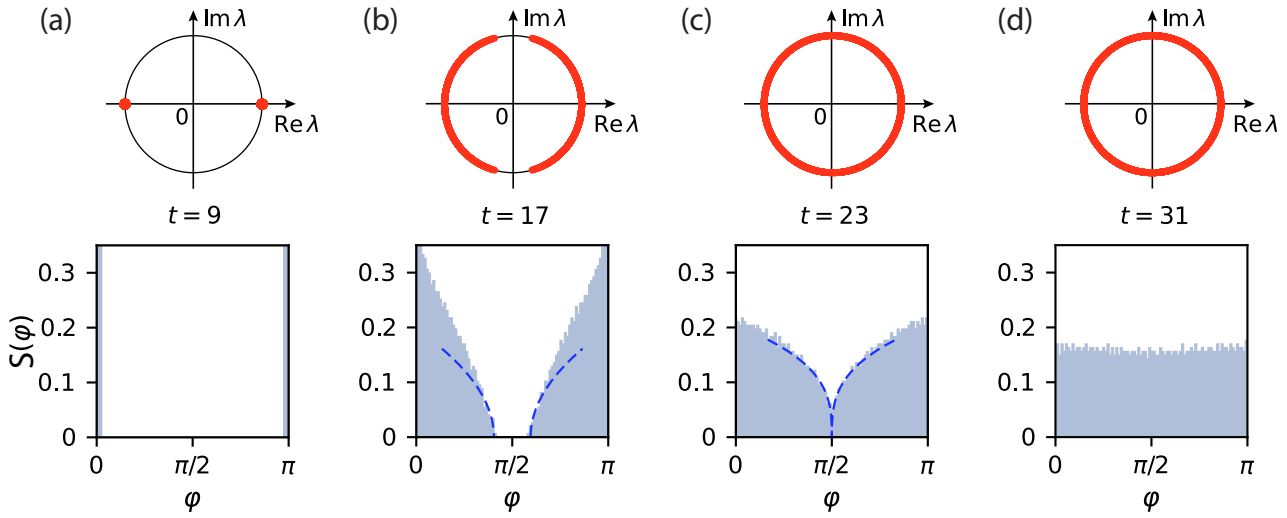


FIG. S55. **Evolution of the correlation operator spectrum in a quantum circuit composed of Haar random two-qubit gates (the system size is $n = 12$).** The behavior of the spectral density $S(\varphi)$ is qualitatively similar to the respective behavior in the random-matrix model (cf. Fig. S52). In panel (**b**), the dashed line shows the dependence $S(\varphi) \propto (|\varphi - \pi/2| - \varphi_{\rm gap})^{1/2} \cdot \Theta(|\varphi - \pi/2| - \varphi_{\rm gap})$ expected from the random-matrix theory [Eqs. (98) and (106)]. The dashed line in panel (**c**) depicts $S(\varphi) \propto |\varphi - \pi/2|^{1/3}$ expected at the critical point (Eq. (107)).

The coveted linear-in-H matrix is constructed in several steps. We begin by noting that the denominator in the expression for the evolution operator U [see Eq. (91)] appears when one inverts an auxiliary matrix

$$\check{L} = \begin{pmatrix} \mathbb{1} & \mathbb{1} \\ H & i\mathbb{1} \end{pmatrix}. \tag{117}$$

In fact, U can be represented as a matrix element of $\check{L}^{-1}$:

$$U = \boldsymbol{v}^T \check{L}^{-1} \boldsymbol{u}, \quad \boldsymbol{u} = \begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix}, \quad \boldsymbol{v} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}. \tag{118}$$

We will denote the auxiliary two-dimensional vector space in which $\check{L}$ acts by a subscript A. Eq. (118) allows us to express the correlation operator as a trace over the auxiliary space:

$$C \equiv \mathrm{Trace}_{\rm A}\big[Z_M \check{\mu}_M (\check{L}^\dagger)^{-1} Z_B \check{\mu}_B \check{L}^{-1}\big], \tag{119}$$

where 2×2 matrices $\check{\mu}_M$ and $\check{\mu}_B$ are given by

$$\check{\mu}_M = \boldsymbol{u}\boldsymbol{u}^T, \qquad \check{\mu}_B = \boldsymbol{v}\boldsymbol{v}^T. \tag{120}$$

Due to the structure of these matrices, not only C but also any power of C can be represented as a trace:

$$C^k \equiv \mathrm{Trace}_{\rm A}\big[\big(Z_M \check{\mu}_M (\check{L}^\dagger)^{-1} Z_B \check{\mu}_B \check{L}^{-1}\big)^k\big], \tag{121}$$

Our key observation is that the object under the trace in Eqs. (119) and (121) emerges naturally when one inverts the matrix

$$\sqrt{\lambda} \begin{pmatrix} \check{0} & \check{L} \\ \check{L}^\dagger & \check{0} \end{pmatrix}_{\rm K} - \begin{pmatrix} Z_M \check{\mu}_M & \check{0} \\ \check{0} & Z_B \check{\mu}_B \end{pmatrix}_{\rm K}. \tag{122}$$

This matrix acts in a further-enlarged Hilbert space; we indicated the new two-dimensional subspace by a subscript K. As follows from Eq. (117), the matrix in Eq. (122) is linear in H. It is this matrix whose inverse will be the main object in the calculation. We denote the inverse by

$$\hat{\mathcal{G}}(\lambda) = \Big[\sqrt{\lambda} \begin{pmatrix} \check{0} & \check{L} \\ \check{L}^\dagger & \check{0} \end{pmatrix}_{\rm K} - \begin{pmatrix} Z_M \check{\mu}_M & \check{0} \\ \check{0} & Z_B \check{\mu}_B \end{pmatrix}_{\rm K}\Big]^{-1}. \tag{123}$$

To establish a direct connection between $\hat{\mathcal{G}}(\lambda)$ and the correlation operator C, it is useful to evaluate

$$\mathrm{Trace}_{\rm K,A}\Big[\begin{pmatrix} Z_M \check{\mu}_M & \check{0} \\ \check{0} & Z_B \check{\mu}_B \end{pmatrix}_{\rm K} \hat{\mathcal{G}}(\lambda)\Big]. \tag{124}$$

Performing a series expansion of $\hat{\mathcal{G}}(\lambda)$ at $\lambda \to \infty$, we obtain

$$\begin{aligned} &\mathrm{Trace}_{\rm K,A}\Big[\begin{pmatrix} Z_M \check{\mu}_M & \check{0} \\ \check{0} & Z_B \check{\mu}_B \end{pmatrix}_{\rm K} \hat{\mathcal{G}}(\lambda)\Big] \\ &= 2\sum_{k=1}^{\infty} \frac{1}{\lambda^k} \mathrm{Trace}_{\rm A}\big[\big(Z_M \check{\mu}_M (\check{L}^\dagger)^{-1} Z_B \check{\mu}_B \check{L}^{-1}\big)^k\big] \\ &= 2\sum_{n=1}^{\infty} \frac{1}{\lambda^k} C^k = 2(\lambda G(\lambda) - \mathbb{1}). \end{aligned} \tag{125}$$

To transition from the second to the third lines, we used representation (121) for the k-th power of the correlation operator; the last equality stems from the definition of the Green's function $G(\lambda) = 1/[\lambda\mathbb{1} - C]$. Finally, we substitute the obtained result in Eq. (116). This leads to

$$\langle \mathrm{Trace}_{\mathrm{H}}[G(\lambda)]\rangle \tag{126}$$
$$= \frac{1}{\lambda}\Big(D_{\mathrm{H}} + \frac{1}{2}\mathrm{Trace}_{\mathrm{K,A,H}}\Big[\begin{pmatrix} Z_M\check{\mu}_M & \check{0} \\ \check{0} & Z_B\check{\mu}_B \end{pmatrix}_{\mathrm{K}} \langle\hat{\mathcal{G}}(\lambda)\rangle\Big]\Big).$$

The main advantage of representing $G(\lambda)$ in terms of $\hat{\mathcal{G}}(\lambda)$ is that the linear dependence of $\hat{\mathcal{G}}^{-1}$ on H. This fact allows us to carry out the ensemble-averaging using the diagrammatic techniques [71]. We explain how this is done in the next paragraph.

Ensemble averaging — In this paragraph, we use the diagrammatic technique to perform the averaging of $\hat{\mathcal{G}}(\lambda)$. A first step is to split $\hat{\mathcal{G}}^{-1}(\lambda)$ into two parts:

$$\hat{\mathcal{G}}^{-1}(\lambda) = \hat{\mathcal{G}}_0^{-1}(\lambda) + \sqrt{\lambda}H\hat{\Pi}_x. \tag{127}$$

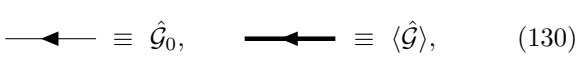
Here $\hat{\mathcal{G}}_0^{-1}(\lambda) \equiv \hat{\mathcal{G}}^{-1}(\lambda)|_{H=0}$ is the bare Green's function and

$$\hat{\Pi}_x = \begin{pmatrix} 0&0&0&0\\ 0&0&1&0\\ 0&1&0&0\\ 0&0&0&0 \end{pmatrix}_{\mathrm{K,A}}. \tag{128}$$

The object that we need to compute to find the spectral density is the average Green's function [see Eq. (126)]

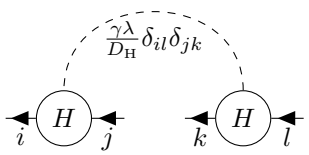
$$\langle\hat{\mathcal{G}}(\lambda)\rangle = \left\langle \frac{1}{\hat{\mathcal{G}}_0^{-1}(\lambda) + \sqrt{\lambda}H\hat{\Pi}_x}\right\rangle. \tag{129}$$

To implement the diagrammatic calculation, we introduce diagrams for the bare and the dressed Green functions:



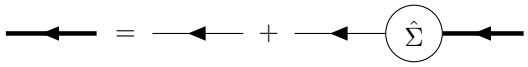
(130)

We will represent averaging over the statistics of Gaussian variables H_{ij} by a dashed line:



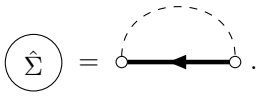
, (131)

see Eq. (90b) [we included the vertex factors of $\sqrt{\lambda}$ into the disorder line]. As usual, the average Green's function satisfies the Dyson equation; in the diagrammatic form, this equation reads:



, (132)

where self-energy $\hat{\Sigma}$ includes all possible pairings of H in the Taylor expansion of Eq. (129). In the limit of large Hilbert space dimension, $D_{\mathrm{H}} \gg 1$, the self-energy can be found with the help of the self-consistent Born approximation [70, 71]:



. (133)

This approximation amounts to a resummation of all diagrams in which the disorder lines do not cross. It is justified by the fact that the crossing diagrams are suppressed by a factor $1/D_{\mathrm{H}} \ll 1$.

Eq. (132) and Eq. (133) are self-consistent: the average Green's function is determined by the self-energy [Eq. (132)], while the self-energy itself depends on $\langle\hat{\mathcal{G}}\rangle$ via Eq. (133). In the next paragraph, we perform the analysis of these equations and use them to derive the self-consistency relation, Eq. (95).

Derivation of the self-consistency relation — Algebraically, the obtained system of self-consistent equations [Eqs. (132) and (133)] has the following form:

$$\langle\hat{\mathcal{G}}\rangle = \hat{\mathcal{G}}_0 + \hat{\mathcal{G}}_0\hat{\Sigma}\langle\hat{\mathcal{G}}\rangle, \tag{134a}$$
$$\hat{\Sigma} = \gamma\lambda\,\mathbb{1}\otimes\hat{\Pi}_x\tfrac{1}{D_{\mathrm{H}}}\mathrm{Trace}_{\mathrm{H}}[\langle\hat{\mathcal{G}}\rangle]\hat{\Pi}_x \tag{134b}$$

Below, we explain how this system can be solved.

To begin with, notice that the presence of matrices $\hat{\Pi}_x$ in Eq. (134b) enforces the form of the self-energy with at most four (out of 16) non-zero components in K and A subspaces:

$$\hat{\Sigma} = \begin{pmatrix} 0&0&0&0\\ 0&\zeta_{11}&\zeta_{12}&0\\ 0&\zeta_{21}&\zeta_{22}&0\\ 0&0&0&0 \end{pmatrix}_{\mathrm{K,A}}. \tag{135}$$

This highlights a special role played by the inner 2×2 block of this matrix, and suggests projecting Eq. (134a) onto this block as well. The projection is implemented by an operator:

$$\hat{\Pi} = \begin{pmatrix} 0&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&0 \end{pmatrix}_{\mathrm{K,A}}. \tag{136}$$

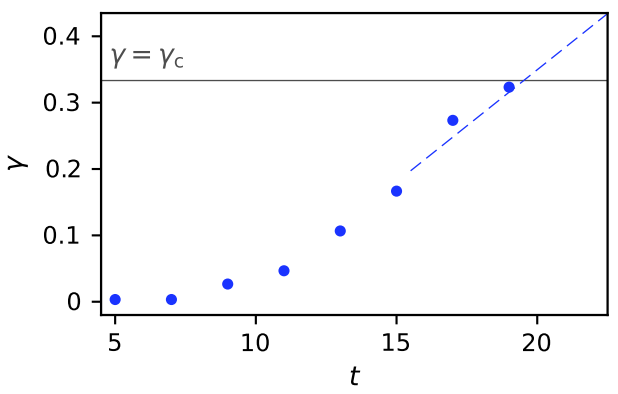


FIG. S56. **Correspondence between parameter γ of the random-matrix theory.** See Eq. (90a), and circuit depth t. To establish the correspondence, we numerically obtain the gap at every t, and then find the value of the random-matrix parameter γ that leads to the same gap. Dashed line shows the extrapolation of $\gamma(t)$ into the domain $\gamma > \gamma_\star$ in which there is no gap.

We introduce a 2×2 matrix g as the inner block of $\langle\hat{\mathcal{G}}\rangle$,

$$\hat{\Pi}\langle\hat{\mathcal{G}}\rangle\hat{\Pi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & g_{11} & g_{12} & 0 \\ 0 & g_{21} & g_{22} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{\mathrm{K,A}}, \tag{137}$$

and similarly g_0 as the inner block of $\hat{\mathcal{G}}_0$. We can represent the projected version of Eq. (134a) in terms of g, g_0, and ζ as

$$g^{-1} = g_0^{-1} - \zeta. \tag{138}$$

Eq. (134b) can also be recast through ζ and g:

$$\zeta = \gamma\lambda\mathbb{1}\otimes\sigma_x \tfrac{1}{D_{\mathrm{H}}}\mathrm{Trace}_{\mathrm{H}}[g]\sigma_x. \tag{139}$$

Here and below, σ_i with $i = x, y, z$ are Pauli matrices in the inner subspace. To finish defining the system of equations for g and ζ, we need to specify g_0^{-1}. A straightforward algebra following from the definition of $\hat{\mathcal{G}}$ (see Eq. (123)) yields

$$g_0^{-1} = \sigma_y\sqrt{\lambda}\,\frac{Z_M Z_B + \lambda\mathbb{1}}{Z_M Z_B - \lambda\mathbb{1}} + \frac{1+\sigma_z}{2}\,\frac{2Z_M\lambda}{Z_M Z_B - \lambda\mathbb{1}} + \frac{1-\sigma_z}{2}\,\frac{2Z_B\lambda}{Z_M Z_B - \lambda\mathbb{1}}. \tag{140}$$

With that, we have all of the components needed to solve Eqs. (138) and (139).

A crucial observation that allows us to make further analytical progress is that the system defined by Eqs. (138) and (139) admits a solution for the self-energy with a very simple matrix structure,

$$\zeta = -i\sqrt{\lambda}\,\varepsilon\,\sigma_y. \tag{141}$$

We introduced a factor of $\sqrt{\lambda}$ here for notational convenience; parameter $\varepsilon \equiv \varepsilon(\lambda)$ depends on λ. The reason why ansatz (141) goes through Eqs. (138) and (139) is that this ansatz preserves the structure of g in which the diagonal terms are proportional to Z_M and Z_B. Because $\mathrm{Trace}_{\mathrm{H}}[Z_{M,B}] = 0$, these terms would nullify on the right-hand side of Eq. (139); only the terms $\propto\sigma_y$ would survive the trace. This is compatible with the left-hand side being $\propto\sigma_y$.

Combining Eqs. (138), (140), and (141), and using them in Eq. (139), we obtain a self-consistency relation for ε:

$$\varepsilon = \frac{\gamma}{2}\sum_{Z=\pm1}\frac{\varepsilon - i\frac{\lambda-Z}{\lambda+Z}}{1-\varepsilon^2 + 2i\varepsilon\frac{\lambda-Z}{\lambda+Z}}. \tag{142}$$

By substituting $\lambda = e^{i\varphi}$ into this equation, we arrive to Eq. (95) of Section IV B 2.

We are now ready to obtain an expression for $\langle\hat{\mathcal{G}}(\lambda)\rangle$, and thus complete the derivation of $\langle\mathrm{Trace}_{\mathrm{H}}[G(\lambda)]\rangle$ [we remind one that the relation between the two Green's functions is given by Eq. (126)]. In fact, $\langle\hat{\mathcal{G}}(\lambda)\rangle$ can be straightforwardly expressed in terms of the already found g_0 and ζ [Eqs. (140)–(142)]. To do that, let us first note an alternative, useful representation of Eq. (134a):

$$\langle\hat{\mathcal{G}}\rangle = \hat{\mathcal{G}}_0 + \hat{\mathcal{G}}_0\hat{\Pi}\hat{\Sigma}(\mathbb{1} - \hat{\mathcal{G}}_0\hat{\Sigma})^{-1}\hat{\Pi}\hat{\mathcal{G}}_0. \tag{143}$$

To insert projectors $\hat{\Pi}$ here, we employed the property $\hat{\Sigma} = \hat{\Pi}\hat{\Sigma} = \hat{\Sigma}\hat{\Pi}$ that follows directly from Eq. (134b). Obviously, the matrix $\hat{\Pi}\hat{\Sigma}(\mathbb{1}-\hat{\mathcal{G}}_0\hat{\Sigma})^{-1}\hat{\Pi}$ differs from zero only in its inner block. Thus, we can represent it as

$$\hat{\Pi}\hat{\Sigma}(\mathbb{1} - \hat{\mathcal{G}}_0\hat{\Sigma})^{-1}\hat{\Pi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \theta_{11} & \theta_{12} & 0 \\ 0 & \theta_{21} & \theta_{22} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{\mathrm{K,A}}. \tag{144}$$

A simple calculation gives

$$\theta = \zeta(\mathbb{1} - g_0\zeta)^{-1} = -i\sqrt{\lambda}\varepsilon\sigma_y(\mathbb{1} + i\sqrt{\lambda}\varepsilon g_0\sigma_y)^{-1}. \tag{145}$$

The use of Eqs. (143)–(145) in Eq. (126) leads to Eq. (94) of Section IV B 2. We have now derived all of the equations needed to compute the spectral density $S(\varphi)$ in the limit $D_{\mathrm{H}} \gg 1$.

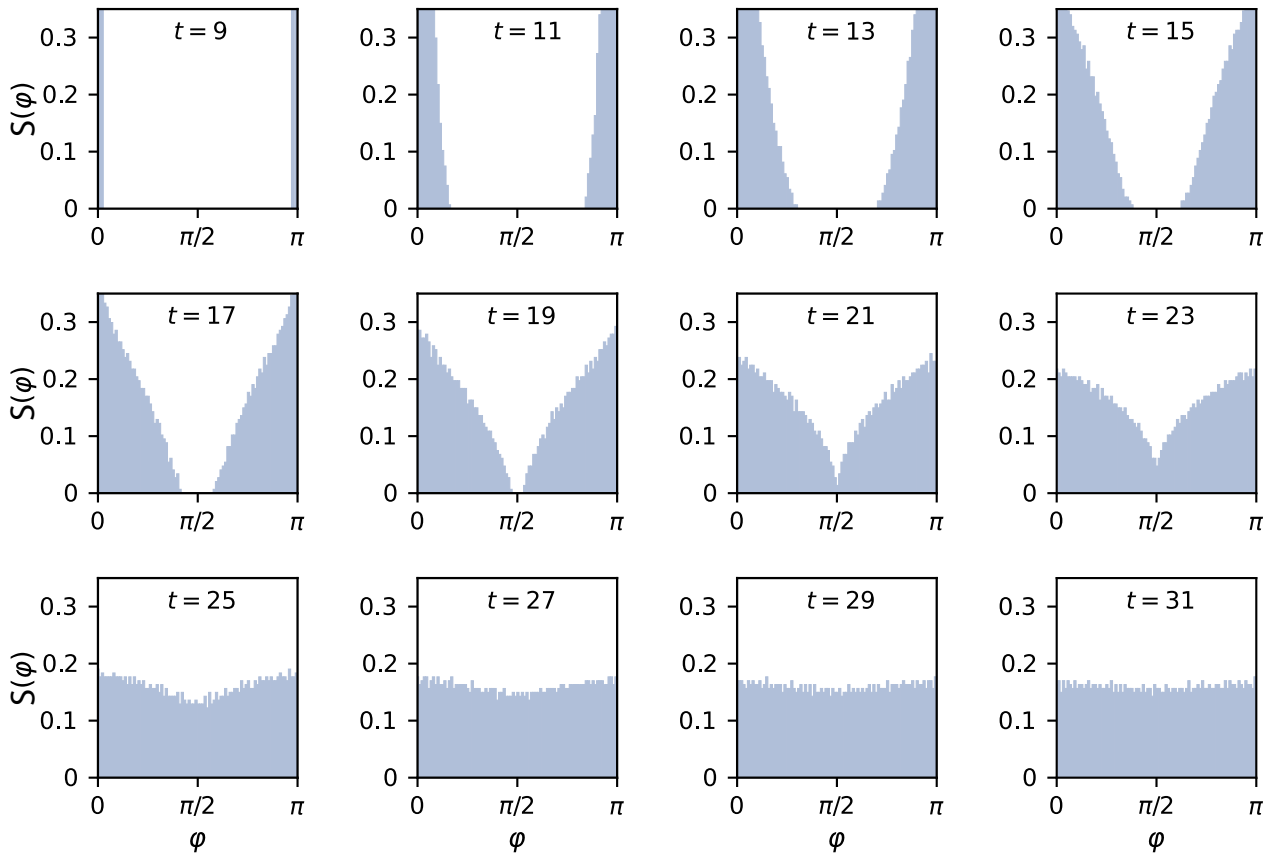


FIG. S57. Full time dependence of the correlation operator spectrum in a quantum circuit composed of Haar random two-qubit gates [the circuit instance is the same as in Fig. S55].

5. *Numerical results on* 1D *random circuits*

Finally, we verify numerically that the phenomenology of spectral phase transition also applies to realistic quantum circuits composed of random, local gates. To this end, we consider a chain of n qubits. As butterfly and measurement operators we use Z_1 and Z_n, respectively. We generate the evolution U by a brickwork circuit composed of two-qubit Haar random gates, and track how the spectrum of $C = Z_n U^\dagger Z_1 U$ changes with the increase of the circuit depth t.

The results of the numerical simulation are presented in Figs. S55 and S57. Even for a modest system size, $n = 12$, they demonstrate a qualitative agreement with the general expectations of Section IV B 1, as well as with predictions of the random-matrix theory model [cf. Fig. S52 and Fig. S55]. To provide the correspondence between γ and t, we match the values of the gap at different times to $\varphi_{\rm gap}(\gamma)$ of the random-matrix theory model, see Fig. S56.

V. TABLE OF SYMBOLS USED IN TEXT

Symbol	Meaning
	General symbols
I, X, Y, Z	Pauli operators
P	Generic Pauli operator $P = \{I, X, Y, Z\}^{\otimes n}$
$\|\psi\rangle$	Quantum state vector
$\|P\rangle\rangle$	Vector in superoperator formalism (corresponding to operator P)
ρ	Density matrix
d	Local Hilbert space dimension
D_H	Hilbert space dimension, $D_{\mathrm{H}} = d^n$
U	Circuit unitary
u_t	Circuit sub-block unitary: $U = \prod_t u_t$.
t_f	Number of cycles or layers in a circuit
Π	Projector
$\langle\cdot\rangle$	Expectation value
$\bar{\cdot}$	Average over circuits
n	Number of qubits
q_m, q_b	Measurement and butterfly qubits
$N_{2\mathrm{D}}$	Number of two-qubit gates within the lightcones of q_m and q_b
p	Probability variable
x, u	Spatial coordinates
t	Time
Trace	Trace, $\mathrm{Trace}[U] = \sum_j U_{jj}$
Tr	Normalized trace, $\mathrm{Tr}[P] = \frac{1}{2^n}\mathrm{Trace}[P]$
$\hat{z}$	Rescaled value of quantity z.
$\widetilde{z}$	Noisy value of quantity z.
	OTOC-related symbols
M	Measurement operator
B	Butterfly operator
$B(t)$	Butterfly operator evolved under unitary; $B(t) = U^\dagger B U$
$C(t)$	Correlation operator $C(t) = MB(t)$
$OTOC^{(1)}$, OTOC	Out-of-time-ordered correlator ($\mathcal{C}^{(2)} = \mathrm{Trace}[\rho MB(t)MB(t)]$)
$\mathcal{C}^{(4)}$, OTOC$^{(2)}$	OTOC squared ($\mathcal{C}^{(4)} = \mathrm{Trace}[\rho MB(t)MB(t)MB(t)MB(t)]$)
$\mathcal{C}^{(2k)}$, OTOC$^{(k)}$	k-th order OTOC ($\mathcal{C}^{(2k)} = \mathrm{Trace}[\rho(MO(t))^{2k}]$)
$\overline{\mathcal{C}}^{(2k)}$	OTOC$^{(k)}$ averaged over a circuit ensemble
L	Loschmidt echo.
$\sigma\left(\mathcal{C}^{(2k)}\right)$	Standard deviation of OTOC$^{(k)}$ over a circuit ensemble
$\mathcal{C}^{(2k)}_{\mathrm{diag}}$	OTOC$^{(k)}$ averaged over a set of inserted Pauli gates
$\mathcal{C}^{(2k)}_{\mathrm{off-diag}}$	Difference between OTOC$^{(k)}$ with and without averaging over Pauli insertion: e.g. $\mathcal{C}^{(2k)}_{\mathrm{off-diag}} = \mathcal{C}^{(2k)} - \mathcal{C}^{(2k)}_{\mathrm{diag}}$.
$\mathcal{C}^{(2k,s)}$	Rescaled OTOC $\mathcal{C}^{(2k,s)} = (\mathcal{C}^{(2k)} - \overline{\mathcal{C}}^{(2k)})/\sigma\left(\mathcal{C}^{(2)}\right)$
$\mathcal{C}^{(4,s)}_{\mathrm{off-diag}}$	Rescaled difference between OTOC^2 values: $\mathcal{C}^{(4,s)}_{\mathrm{off-diag}} = \left(\mathcal{C}^{(4)}_{\mathrm{off-diag}} - \overline{\mathcal{C}^{(4)}}_{\mathrm{off-diag}}\right)/\sigma(\mathcal{C}^{(4)}_{\mathrm{off-diag}})$.
SNR	Signal-to-noise ratio
$\mathcal{C}^{(2k)}_{<\mathrm{method}>}$	<method> approximation of the OTOC$^{(k)}$.
$\mathrm{SNR}_{<\mathrm{method}>}$	SNR of <method> approximation of the OTOC$^{(k)}$.
v_B	Butterfly velocity
D	Front diffusion constant
$b_j(t)$, $b_\alpha(t)$	coefficient of butterfly operator in some basis decomposition
α, β	Pauli path.

Continued on next page

Symbol	Meaning
	Experimental details
T_1, T_2 T_ϕ	Decoherence times
τ	Gate time
$f_{\max}$, f_{10}	Maximum allowed qubit frequency, idle frequency
G_{ij}	iSWAP-like gate
τ_r	Resonance time during G_{ij}
g, $g_{\max}$	Coupling strength, maximum coupling strength during G_{ij}.
	Noise and error mitigation
ϵ_X	Error probability per event X (e.g. a gate).
Γ_{12}	Leakage rate.
F_{noise}	Noise damping factor
K	Number of shots (experiment repetitions)
ϵ_{stats}	Statistical shot noise
ϵ_{sys}	Systematic error
ϵ_{tot}	Total error
SNR_{sys}	Systematic error contribution to the SNR.
	Classical computation details
FLOP	Floating point operations.
FLOPS	Floating point operations per second.
W	Max tensor width during contraction.
ρ_A	Reduced density operator on subsystem A.
N_{projs}	Number of projections in final step of tensor network contraction
$\text{SNR}_{\text{projs}}, a_{\text{SNR}}, b_{\text{SNR}}$	Coefficients of convergence of tensor network contraction with projection count.
$\mathcal{C}$	Cache of Pauli paths in cached-MC algorithm.
b_α	operator amplitude in cached-MC algorithm.
Λ	Pauli transfer matrix.
c	Exponent on iSWAP gate when considering other types of random circuits.

[1] Mi, X. *et al.* Information scrambling in quantum circuits. *Science* **374**, 1479–1483 (2021).

[2] Morvan, A. *et al.* Phase transitions in random circuit sampling. *Nature* **634**, 328–333 (2024).

[3] Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).

[4] Westerhout, T., Astrakhantsev, N., Tikhonov, K. S., Katsnelson, M. I. & Bagrov, A. A. Generalization properties of neural network approximations to frustrated magnet ground states. *Nat. Commun.* **11** (2020).

[5] Lin, S.-H. & Pollmann, F. Scaling of neural-network quantum states for time evolution. *physica status solidi (b)* **259** (2022).

[6] Hémery, K., Pollmann, F. & Luitz, D. J. Matrix product states approaches to operator spreading in ergodic quantum systems. *Phys. Rev. B* **100** (2019).

[7] Östlund, S. & Rommer, S. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.* **75**, 3537–3540 (1995).

[8] Vidal, G. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.* **91** (2003).

[9] Kechedzhi, K. *et al.* Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments. *Future Gener. Comput. Syst.* **153**, 431–441 (2024).

[10] Begušić, T. & Chan, G. K. Real-time operator evolution in two and three dimensions via sparse pauli dynamics. *arXiv:2409.03097* (2024).

[11] Angrisani, A., Mele, A. A., Rudolph, M. S., Cerezo, M. & Holmes, Z. Simulating quantum circuits with arbitrary local noise using pauli propagation. *arXiv:2501.13101* (2025).

[12] Acharya, R. *et al.* Quantum error correction below the surface code threshold. *Nature* **638**, 920–926 (2025).

[13] Barends, R. *et al.* Diabatic gates for frequency-tunable superconducting qubits. *Phys. Rev. Lett.* **123**, 210501 (2019).

[14] Neill, C. *et al.* Accurately computing the electronic properties of a quantum ring. *Nature* **594**, 508–512 (2021).

[15] Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).

[16] Wu, Y. *et al.* Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* **127**, 180501 (2021).

[17] Zhu, Q. *et al.* Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Sci. Bull.* **67**, 240–245 (2022).

[18] DeCross, M. *et al.* The computational power of random quantum circuits in arbitrary geometries. *arXiv:2406.02501* (2024).

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

[20] Huang, C. *et al.* Classical simulation of quantum supremacy circuits. *arXiv:2005.06787* (2020).

[21] Pan, F., Chen, K. & Zhang, P. Solving the sampling problem of the sycamore quantum circuits. *Phys. Rev. Lett.* **129**, 090502 (2022).

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

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

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

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

[26] Villalonga, B. *et al.* A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. *npj Quantum Inf.* **5**, 86 (2019).

[27] Guillon, T., Baker, D. B. & Conradi, M. S. New, compensated carr-purcell sequences. *J. Magn. Reson. (1969)* **89**, 479–484 (1990).

[28] Claeys, P. W. & Lamacraft, A. Maximum velocity quantum circuits. *Phys. Rev. Research* **2**, 033032 (2020).

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

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

[31] Acharya, R. *et al.* Suppressing quantum errors by scaling a surface code logical qubit. *Nature* **614**, 676–681 (2023).

[32] Begušić, T. & Chan, G. K. Fast classical simulation of evidence for the utility of quantum computing before fault tolerance. *arXiv:2306.16372* (2023).

[33] Tindall, J., Fishman, M., Stoudenmire, E. M. & Sels, D. Efficient tensor network simulation of ibm's eagle kicked ising experiment. *PRX Quantum* **5**, 010308 (2024).

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

[35] Biamonte, J. & Bergholm, V. Tensor networks in a nutshell. *arXiv:1708.00006* (2017).

[36] Boixo, S., Isakov, S. V., Smelyanskiy, V. N. & Neven, H. Simulation of low-depth quantum circuits as complex undirected graphical models. *https://arxiv.org/abs/1712.05384* (2017).

[37] Arnborg, S., Corneil, D. G. & Proskurowski, A. Complexity of finding embeddings in ak-tree. *SIAM J. Algebr. Discret. Methods* **8**, 277–284 (1987).

[38] Bravyi, S., Gosset, D. & Liu, Y. How to simulate quantum measurement without computing marginals. *Phys. Rev. Lett.* **128**, 220503 (2022).

[39] Troyer, M. & Wiese, U.-J. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Phys. Rev. Lett.* **94**, 170201 (2005).

[40] Patra, S., Jahromi, S. S., Singh, S. & Orús, R. Efficient tensor network simulation of ibm's largest quantum processors. *Phys. Rev. Res.* **6**, 013326 (2024).

[41] Mauron, L. & Carleo, G. Challenging the quantum advantage frontier with large-scale classical simulations of annealing dynamics. *arXiv:2503.08247* (2025).

[42] Carleo, G., Bauer, B. & Troyer, M. Simulating adiabatic quantum computation with a variational approach. *arXiv:2403.05147* (2024).

[43] Schmitt, M., Rams, M. M., Dziarmaga, J., Heyl, M. & Zurek, W. H. Quantum phase transition dynamics in the two-dimensional transverse-field ising model. *Science Advances* **8** (2022). URL http://dx.doi.org/10.1126/sciadv.abl6850.

[44] Sinibaldi, A., Hendry, D., Vicentini, F. & Carleo, G. Time-dependent neural galerkin method for quantum dynamics (2025). URL https://arxiv.org/abs/2412.11778. 2412.11778.

[45] Schmitt, M. & Heyl, M. Quantum many-body dynamics in two dimensions with artificial neural networks. *Physical Review Letters* **125** (2020). URL http://dx.doi.org/10.1103/PhysRevLett.125.100503.

[46] de Walle, A. V., Schmitt, M. & Bohrdt, A. Many-body dynamics with explicitly time-dependent neural quantum states (2024). URL https://arxiv.org/abs/2412.11830. 2412.11830.

[47] Chen, A., Naik, V. D. & Heyl, M. Convolutional transformer wave functions (2025). URL https://arxiv.org/abs/2503.10462. 2503.10462.

[48] Sorella, S. Green function monte carlo with stochastic reconfiguration. *Physical Review Letters* **80**, 4558–4561 (1998). URL http://dx.doi.org/10.1103/PhysRevLett.80.4558.

[49] Kingma, D. P. & Ba, J. Adam: A method for stochastic

optimization. *arXiv:1412.6980* (2014).

[50] Döschl, F., Palm, F. A., Lange, H., Grusdt, F. & Bohrdt, A. Neural network quantum states for the interacting hofstadter model with higher local occupations and long-range interactions (2024). URL https://arxiv.org/abs/2405.04472. 2405.04472.

[51] Sharir, O., Shashua, A. & Carleo, G. Neural tensor contractions and the expressive power of deep neural quantum states. *Physical Review B* **106** (2022). URL http://dx.doi.org/10.1103/PhysRevB.106.205136.

[52] Levine, Y., Sharir, O., Cohen, N. & Shashua, A. Quantum entanglement in deep learning architectures. *Physical Review Letters* **122** (2019). URL http://dx.doi.org/10.1103/PhysRevLett.122.065301.

[53] Deng, D.-L., Li, X. & Das Sarma, S. Quantum entanglement in neural network states. *Physical Review X* **7** (2017). URL http://dx.doi.org/10.1103/PhysRevX.7.021021.

[54] Denis, Z., Sinibaldi, A. & Carleo, G. Comment on "can neural quantum states learn volume-law ground states?". *Physical Review Letters* **134** (2025). URL http://dx.doi.org/10.1103/PhysRevLett.134.079701.

[55] Passetti, G. *et al.* Can neural quantum states learn volume-law ground states? *Phys. Rev. Lett.* **131**, 036502 (2023).

[56] Czarnik, P., Dziarmaga, J. & Corboz, P. Time evolution of an infinite projected entangled pair state: An efficient algorithm. *Physical Review B* **99** (2019). URL http://dx.doi.org/10.1103/PhysRevB.99.035115.

[57] Haghshenas, R. *et al.* Digital quantum magnetism at the frontier of classical simulations (2025). URL https://arxiv.org/abs/2503.20870. 2503.20870.

[58] Liu, Z.-W. & Winter, A. Many-body quantum magic. *PRX Quantum* **3** (2022). URL http://dx.doi.org/10.1103/PRXQuantum.3.020333.

[59] Garcia, R. J., Bu, K. & Jaffe, A. Resource theory of quantum scrambling. *Proceedings of the National Academy of Sciences* **120** (2023). URL http://dx.doi.org/10.1073/pnas.2217031120.

[60] Liu, Q. & Ma, W. The epochal sawtooth effect: Unveiling training loss oscillations in adam and other optimizers (2025). URL https://arxiv.org/abs/2410.10056. 2410.10056.

[61] Khemani, V., Vishwanath, A. & Huse, D. A. Operator spreading and the emergence of dissipative hydrodynamics under unitary evolution with conservation laws. *Phys. Rev. X* **8**, 031057 (2018).

[62] Fisher, M. P. A., Khemani, V., Nahum, A. & Vijay, S. Random quantum circuits. *Annu. Rev. Condens. Matter Phys.* **14**, 335–379 (2023).

[63] Bertini, B. & Piroli, L. Scrambling in random unitary circuits: Exact results. *Phys. Rev. B.* **102**, 064305 (2020).

[64] Diaconis, P. Group representations in probability and statistics. *Lect. Notes-Monogr. Ser.* **11**, i–192 (1988).

[65] Collins, B., Matsumoto, S. & Novak, J. The weingarten calculus. *Notices Am. Math. Soc.* **69**, 734 (2022).

[66] Weingarten, D. Asymptotic behavior of group integrals in the limit of infinite rank. *J. Math. Phys.* **19**, 999–1001 (1978).

[67] Zhou, T. & Nahum, A. Emergent statistical mechanics of entanglement in random unitary circuits. *Phys. Rev. B* **99**, 174205 (2019).

[68] Kardar, M., Parisi, G. & Zhang, Y. C. Dynamic scaling of growing interfaces. *Phys. Rev. Lett.* **56**, 889–892 (1986).

[69] Vidal, G. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.* **93**, 040502 (2004).

[70] Khindanov, A., Aleiner, I. L., Faoro, L. & Ioffe, L. B. Observable measurement-induced transitions. *arXiv:2410.09353* (2024).

[71] Abrikosov, A. A., Gorkov, L. P. & Dzyaloshinski, I. E. *Methods of Quantum Field Theory in Statistical Physics.* (Prentice Hall, Englewood Cliffs, N. J., 1963), 1st edn.