Efficient Classical Simulation of Random Shallow 2D Quantum Circuits

John C. Napp[1], Rolando L. La Placa,[1] Alexander M. Dalzell[2], Fernando G. S. L. Brandão,[2,3] and Aram W. Harrow[1]

[1]*Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*
[2]*Institute for Quantum Information and Matter, Caltech, Pasadena, California 91125, USA*
[3]*AWS Center for Quantum Computing, Pasadena, California 91125, USA*

A central question of quantum computing is determining the source of the advantage of quantum computation over classical computation. Even though simulating quantum dynamics on a classical computer is thought to require exponential overhead in the worst case, efficient simulations are known to exist in several special cases. It was widely assumed that these easy-to-simulate cases as well as any yet-undiscovered ones could be avoided by choosing a quantum circuit at random. We prove that this intuition is false by showing that certain families of constant-depth, 2D random circuits can be approximately simulated on a classical computer in time only linear in the number of qubits and gates, even though the same families are capable of universal quantum computation and are hard to exactly simulate in the worst case (under standard hardness assumptions). While our proof applies to specific random circuit families, we demonstrate numerically that typical instances of more general families of sufficiently shallow constant-depth 2D random circuits are also efficiently simulable. We propose two classical simulation algorithms. One is based on first simulating spatially local regions which are then "stitched" together via recovery maps. The other reduces the 2D simulation problem to a problem of simulating a form of 1D dynamics consisting of alternating rounds of random local unitaries and weak measurements. Similar processes have recently been the subject of an intensive research focus, which has observed that the dynamics generally undergo a phase transition from a low-entanglement (and efficient-to-simulate) regime to a high-entanglement (and inefficient-to-simulate) regime as measurement strength is varied. Via a mapping from random quantum circuits to classical statistical mechanical models, we give analytical evidence that a similar computational phase transition occurs for both of our algorithms as parameters of the circuit architecture like the local Hilbert space dimension and circuit depth are varied and, additionally, that the effective 1D dynamics corresponding to sufficiently shallow random quantum circuits falls within the efficient-to-simulate regime. Implementing the latter algorithm for the depth-3 "brickwork" architecture, for which exact simulation is hard, we find that a laptop could simulate typical instances on a 409×409 grid with a total variation distance error less than 0.01 in approximately one minute per sample, a task intractable for previously known circuit simulation algorithms. Numerical results support our analytic evidence that the algorithm is asymptotically efficient.

I. MOTIVATION

Simulating quantum systems on classical computers requires effort that scales exponentially with the size of the quantum system, at least for a general-purpose simulation. This was the original motivation of the field of quantum computing, since it shows that time evolution (natural or controlled) of quantum systems can perform tasks that are intractable for classical computers. However, many special cases are known where quantum time evolution *can* be efficiently simulated, including limited entanglement or interaction, free fermions, and Clifford circuits. Understanding the boundary between classically simulable and intractable gets at a crucial question: What makes quantum computing powerful?

There are two main ways to answer this question concretely. We can find more classes of easy-to-simulate quantum dynamics, or we can find evidence that other classes of quantum dynamics are hard for classical computers to simulate. The latter approach is related to the goal of *quantum computational supremacy* [1], which involves finding a well-defined computational task with evidence for classical intractability (usually based on a plausible

conjecture from complexity theory), then actually performing the task on quantum hardware, and verifying the result. Achieving this feat (as has been claimed by Refs. [2–4]) is the computational analog of a Bell inequality violation: Theoretically, it would merely confirm orthodox interpretations of quantum mechanics, but practically it would be a milestone in our ability to coherently control quantum systems.

A leading proposal for demonstrating quantum computational supremacy is *random circuit sampling* (RCS), meaning that the quantum computer applies random unitary gates and then measures all the qubits. This was used by Google [2], and, as we discuss below, it is a plausible candidate for an intractable class of dynamics. Indeed, the previously known examples of efficiently simulable quantum dynamics are all in some ways special: using only Clifford gates or only nonentangling gates, for example. So it would be reasonable to assume that random gates would be the best way to avoid any known or unknown structure in the circuits that would facilitate simulation.

Our main contribution is to show that RCS becomes easy to simulate at low enough circuit depth and local dimension. We do this by developing classical algorithms for RCS that are efficient (polynomial time) in some settings in which all previously known classical algorithms are inefficient (exponential time). Moreover, in these regimes no efficient classical sampling algorithms are possible for arbitrary (i.e., nonrandom) circuits, assuming standard complexity theoretic assumptions (specifically, the "noncollapse of the polynomial hierarchy"). Our results thus show that, for natural problems, random instances can be much easier than a worst-case analysis would suggest. On the other hand, we also find evidence that our algorithms exhibit computational phase transitions into inefficient regimes when certain parameters of the circuits are tuned.

While studying the classical simulability of any nontrivial class of quantum circuits is inherently interesting for understanding the boundary between classical and quantum computation, in the remainder of this section, we give high-level motivation for why studying random quantum circuits, in particular, is *especially* important for the research program of demonstrating quantum computational supremacy.

A. Random circuit sampling

Before specifying the RCS task that we are concerned with in this work, it is worthwhile to begin with a discussion of quantum computational supremacy proposals more generally. Modern proposals usually involve *sampling problems*. A sampling problem asks, given an input string x, for one to generate a sample according to some probability distribution $\mathcal{D}_x$ defined as a function of x. This generalizes the usual task of computing a deterministic function of x.

As a natural example, suppose x encodes the description of a quantum circuit, and $\mathcal{D}_x$ is the *output distribution*

associated with circuit x, defined to be the distribution over output strings induced by measuring each qubit in the computational basis after applying the quantum circuit described by x. Then the following is an example of a sampling problem: Given a quantum circuit description x, generate a sample from $\mathcal{D}_x$. There is a trivial quantum algorithm which executes this task: Just implement the circuit x, measure all qubits, and report the measurement result.

The task of sampling from $\mathcal{D}_x$ appears to be much more challenging for a classical computer, however, as a straightforward classical simulation of the circuit x would incur an exponential overhead. Indeed, it was shown by Terhal and DiVincenzo in 2002 [5] that there does not exist any efficient classical algorithm for sampling from $\mathcal{D}_x$ in general unless the polynomial hierarchy collapses to the third level, a consequence considered to be unlikely. A stronger implication of their work, which is relevant to the setting of constant-depth 2D circuits studied in this paper, is that this hardness result applies even to depth-3 2D quantum circuits. (Note that Ref. [5] defines "depth" using a different convention than ours; we count only the unitary gates and not the final layer of measurements.)

While the Terhal-DiVincenzo result implies that there likely does not exist an efficient classical algorithm that samples from the output distribution of an arbitrary depth-3 quantum circuit, there are obstacles in turning this into a proposal for demonstrating quantum computational supremacy. One is that, while this result implies that there likely cannot exist a classical algorithm for simulating an *arbitrary* circuit instance, it cannot guarantee that a specific given instance is hard to simulate. In other words, Ref. [5] is concerned with worst-case complexity; any efficient classical algorithm must fail on *some* circuit instance but, in principle, could succeed for the vast majority of instances. Another obstacle is that their result implies that it is hard to *exactly* sample from $\mathcal{D}_x$, but it is most natural to additionally require the hardness of sampling from any approximate distribution $\tilde{\mathcal{D}}_x$ which is close to the true distribution in some meaningful sense. Approximate sampling is natural because (a) imperfect quantum computers produce approximate samples, and (b) verifiers that test only a limited (say, polynomial) number of samples, are not able to reliably distinguish approximate samples from exact samples.

To circumvent these obstacles, one usually goes beyond Ref. [5] by conjecturing that a certain family of quantum circuits' output distributions are typically hard for classical computers to even *approximately* sample from in the *average case*. While the classical hardness result of Ref. [5] relies only on the noncollapse of the polynomial hierarchy (a weak conjecture), much stronger conjectures are required for such approximate, average-case hardness statements. The three most well-known supremacy proposals of this form are based on linear-optical networks [6], nstantaneous quantum polynomial (IQP) time circuits [7–9], and random circuits [2,10,11].

In this paper, we study *random quantum circuits* (formally defined subsequently). In addition to their proposed utility in demonstrating quantum computational supremacy, random quantum circuits also find a plethora of applications in quantum information (e.g., scrambling and quantum pseudorandomness) and physics (e.g., operator spreading and entanglement growth under chaotic quantum dynamics). The quantum computational supremacy proposal based on random circuits is known as *random circuit sampling* and is the task of approximately sampling from the output distribution of a random quantum circuit, with high probability of success over circuit instance. The idea for a supremacy proposal based on RCS originates from an email thread between several quantum computing researchers in 2015, in which the participants came to the conclusion that, for the purpose of performing a supremacy experiment on a very near-term quantum device, an experiment based on random circuits was better than the alternative possibilities both from an engineering standpoint [12] and due to the considerations from complexity theory discussed in Sec. II A. In 2019, Google announced a demonstration of quantum computational supremacy on the basis of implementing a RCS experiment using a 2D array of 53 superconducting qubits with nearest-neighbor interactions [2].

RCS was an extremely natural candidate for a supremacy proposal. Experimentally, it was amenable to implementation in the very near term. Theoretically, there was intuition supporting the belief that random quantum circuits should be hard for classical computers to simulate. The Terhal-DiVincenzo result already implied that exactly simulating random quantum circuits should be classically hard in the worst case; that is, no classical algorithm should work on *all* instances. Essentially, their result needed to be extended from an "exact, worst-case" result to an "approximate, average-case" result. This extension indeed felt plausible, because (in the noiseless setting) efficient classical simulation algorithms are generally known only for classes of quantum circuits that are highly structured. Therefore, one might expect that randomizing the gates of a quantum circuit would wash out any structure that would allow for an efficient classical simulation and make random gates essentially the hardest possible to classically simulate.

After the idea of demonstrating quantum computational supremacy via RCS was introduced, a number of groups studied the computational complexity of RCS, collecting evidence (detailed in the following section) that RCS should be hard for a classical computer. Nonetheless, we describe two classical algorithms for RCS for 2D random circuits and give evidence for their efficiency on circuits of sufficiently shallow depth. In fact, these algorithms are efficient for some classes of quantum circuits for which much of this prior evidence of classical hardness is applicable. Arguably, in demonstrating that there exist classes of random circuits for which RCS is classically

tractable yet some of these seeming pieces of evidence of classical hardness apply, the previous intuition and theoretical underpinning for the classical hardness of RCS— and, by extension, Google's quantum supremacy claim—is weakened. On the other hand, despite weakening some of the previous reasons in believing in the classical hardness of RCS, this paper could also be viewed as contributing new evidence in *support* of the hardness of simulating sufficiently *deep* 2D random quantum circuits. This is because the two novel classical simulation algorithms that we introduce appear to experience computational phase transitions as the circuit depth is increased. Roughly speaking, for a particular 2D circuit architecture, we find evidence that there is a critical constant depth d^* such that, in performing random circuit sampling on a $\sqrt{n} \times \sqrt{n}$ square array of qubits of depth d, our algorithms run in time poly(n) if $d < d^*$ and run in time approximately $\exp(n^{O(1)})$ if $d > d^*$. We find a similar phase transition as a function of the local Hilbert space dimension if more general *qudits* of dimension possibly greater than two are used instead of qubits. We hope that, in addition to showing that RCS may be efficiently solved classically for sufficiently shallow 2D circuits, this work may also help initiate a deeper line of inquiry into how the hardness of classically simulating random quantum circuits depends on architecture parameters such as the depth, local dimension, and number of spatial dimensions.

II. OVERVIEW OF CONTRIBUTIONS

As discussed in the previous section, a fundamental question in computer science and physics is to understand where the boundary between classically intractable and classically simulable quantum systems or quantum circuits lies. A more specific question within the context of quantum computational supremacy is to understand what types of quantum gate sequences are hardest to classically simulate. So far, our answers to these questions have been informal or incomplete. On the simulation side, Markov and Shi [13] show that a quantum circuit could be classically simulated by contracting a tensor network with cost exponential in the tree width of the graph induced by the circuit. (Tree width is a measure of how far from a tree a graph is; it is 1 for a tree and approximately L^{D-1} for a D-dimensional lattice with side length L.) When applied to n qubits in a line running a circuit with depth d, the simulation cost of this algorithm is $\exp\{O[\min(n, d)]\}$. More generally, we could consider $n = L_1 L_2$ qubits arranged in an $L_1 \times L_2$ grid running for depth d, in which case the simulation cost would be

$$\exp\{O[\min(L_1 L_2, L_1 d, L_2 d)]\}. \tag{1}$$

In other words, we can think of the computation as taking up a space-time volume of $L_1 \times L_2 \times d$ and the simulation cost is dominated by the size of the smallest cut bisecting

this volume. An exception is for depth $d = 1$ or $d = 2$, which have simple exact simulations [5]. Some restricted classes such as stabilizer circuits [14] or one-dimensional systems that are sufficiently unentangled [15–17] may also be simulated efficiently. However, the conventional wisdom has been that, in general, for 2D circuits with $d \geq 3$, the simulation cost scales as Eq. (1).

These considerations led IBM to propose the benchmark of "quantum volume" [18] which in our setting is $\exp[\sqrt{d \min(L_1, L_2)}]$; this does not exactly coincide with Eq. (1) but qualitatively captures a similar phenomenon. The idea of quantum volume is to compare quantum computers with possibly different architectures by evaluating their performance on a simple benchmark. This benchmark task is to perform n layers of random two-qubit gates on n qubits, and being able to perform this with $\lesssim 1$ expected gate errors corresponds to a quantum volume of $\exp(n)$. [19] Google's quantum computing group has also proposed random unitary circuits as a benchmark task for quantum computers [10]. While their main goal has been quantum computational supremacy [2,21], random circuits could also be used to diagnose errors including those that go beyond single-qubit error models by more fully exploring the configuration space of the system [18].

These proposals from industry reflect a rough consensus that simulating a 2D random quantum circuit should be nearly as hard as exactly simulating an arbitrary circuit with the same architecture or, in other words, that random circuit simulation is nearly as hard as the *worst case*, given our current state of knowledge.

To the contrary, we prove (assuming standard complexity-theoretic conjectures) that, for a certain family of constant-depth architectures, classical simulation of typical instances with a small allowed error is easy, despite worst-case simulation being hard (by which we mean it is classically intractable to simulate an arbitrary random circuit realization with an arbitrarily small error). For these architectures, we show that a certain algorithm exploiting the randomness of the gates and the allowed small simulation error can run much more quickly than the scaling in Eq. (1), running in time $O(L_1 L_2)$. While our proof is architecture specific, we give numerical and analytical evidence that, for sufficiently low constant values of d, the algorithm remains efficient more generally. The intuitive reason for this is that the simulation of 2D shallow random circuits can be reduced to the simulation of a form of effective 1D dynamics which includes random local unitaries and weak measurements. The measurements cause the 1D process to generate much less entanglement than it could in the worst case, making efficient simulation possible. Such dynamics consisting of random local gates with interspersed measurements has, in fact, recently become the subject of an intensive research focus [22–49], and our simulation algorithm can be viewed as an application of this line of work. Furthermore, the

measurement-strength-driven entanglement phase transitions observed in these processes are closely related to the computational phase transition we observe for our algorithms.

A. Evidence from prior literature that simulating random circuits is hard

Before discussing our results in greater detail, we briefly review the main technical arguments for the prevailing belief that random circuit simulation should be nearly as hard as the worst case.

1. Evidence from complexity theory

A long line of work has shown that it is worst-case hard to either sample from the output distributions of quantum circuits or compute their output probabilities with an exponentially small error [1,5,6,8,9,50,51]. While the requirements of worst-case and near-exact simulation are rather strong, these results do apply to any quantum circuit family that becomes universal once postselection [50] is allowed, thereby including noninteracting bosons [6] and 2D depth-3 circuits [5]. The hardness results are also based on the widely believed conjecture that the polynomial hierarchy (PH) is infinite or, more precisely, that approximate counting is weaker than exact counting. Since these results naturally yield worst-case hardness, they do not obviously imply that random circuits should be hard. In some cases, additional conjectures can be made to extend the hardness results to some form of average-case hardness (as well as ruling out approximate simulations) [6,9,11], but these conjectures have not received widespread scrutiny. Besides stronger conjectures, these hardness results usually require that the quantum circuits have an "anticoncentration" property, meaning roughly that their outputs are not too far from the uniform distribution [52]. While random circuits are certainly not the only route to anticoncentration (applying a Hadamard gate to each qubit of $|0\rangle^{\otimes n}$ would do), they are a natural way to combine anticoncentration with an absence of any obvious structure (e.g., Clifford gates) that might admit a simple simulation (however, note that constant-depth random quantum circuits do not have the anticoncentration property [53]). Furthermore, a line of work beginning with Ref. [54] (see Refs. [55–57] for subsequent improvements) has established that random circuit simulation admits a worst-to-average-case reduction for the computation of output probabilities. In particular, the ability to near-exactly compute the probability of some output string for a $1 - 1/\text{poly}(n)$ fraction of Haar-random circuit instances on some architecture is essentially as hard as computing output probabilities for an arbitrary circuit instance with this architecture, which is known to be #P-hard even for certain 2D depth-3 architectures.

2. Near-maximal entanglement in random circuits

Haar-random states on n qudits are nearly maximally entangled across all cuts simultaneously [58,59]. Random quantum circuits on $L \times L \times \cdots$ arrays of qudits achieve similar near-maximal entanglement across all possible cuts once the depth is at least approximately L [52,60], and, before this time, the entanglement often spreads "ballistically" [61,62]. Random tensor networks with large bond dimension nearly obey a min-flow and max-cut-type theorem [63,64], again meaning that they achieve nearly maximal values of an entanglementlike quantity. These results suggest that, when running algorithms based on tensor contraction, random gates should be nearly the hardest possible gates to simulate.

3. Absence of algorithms taking advantage of random inputs

There are not many algorithmic techniques known that simulate random circuits more easily than worst-case circuits. There are a handful of exceptions. In the presence of any constant rate of noise, random circuits [65,66], IQP circuits [51], and (for photon loss) boson sampling [67,68] can be efficiently simulated. These results can also be viewed as due to the fact that fault-tolerant quantum computing is not a generic phenomenon and requires structured circuits to achieve (see Ref. [51] for a discussion in the context of IQP). Permanents of random matrices whose entries have small nonzero mean can be approximated efficiently [69], while the case of boson sampling corresponds to entries with zero mean and the approach of Ref. [69] is known to fail there. A heuristic approximate simulation algorithm based on tensor network contraction [70] was recently proposed and applied to random circuits, although for this algorithm it is unclear how the approximations made are related to the overall simulation error incurred (in contrast, our algorithm based on matrix product states can bound the overall simulation error it is making, even when comparison with exact simulation is not feasible). In practice, evidence for a hardness conjecture often is no more than the absence of algorithms. Indeed, while some approximation algorithms are known for estimating output probabilities of constant-depth circuits [71], IQP circuits [7], and boson sampling [6] up to additive error δ in time poly$(n, 1/\delta)$, these are not very helpful for random circuits where typical output probabilities are approximately 2^{-n}.

Despite the above intuitive arguments for why the simulation of uniformly random circuits should be nearly as hard as the worst case, we (i) prove that there exist architectures for which this is not the case and (ii) give evidence that this result is not architecture specific but is rather a general property of sufficiently shallow random circuits. To this end, we propose and implement a simulation algorithm based on a 2D-to-1D mapping in conjunction with tensor network methods. In Appendix B of Supplemental Material [72], we introduce and study a second simulation algorithm (referred to as Patching) based on locally simulating spatially disconnected regions which are then "stitched" together. The performance of both algorithms is related to certain entropic quantities.

We also give evidence of computational phase transitions for our proposed simulation algorithms driven by circuit depth and qudit dimension. Previously, it was known that phase transitions between classical and quantum computation exist as a function of the noise parameter in conventional quantum computation [73–79] as well as in measurement-based quantum computing (MBQC) [80,81]. In the noiseless setting, besides the gap between depth-2 and depth-3 circuits [5], a computational phase transition as a function of the rate of qubit loss during the preparation of a resource state for MBQC [82] and (under additional assumptions) as a function of the duration of time evolution for simulating dynamics generated by quadratic bosonic Hamiltonians [83,84] was also known.

For the case of constant depth, there have been some quantum computational supremacy proposals that do not use uniformly random circuits, mostly based on the measurement-based quantum computing model [85]. This means first preparing a cluster state and then measuring it in mostly equatorial bases or, equivalently, performing $e^{i\theta Z}$ for various angles θ and then measuring in the X basis. This is far from performing uniformly random nearest-neighbor gates up to the same depth and then measuring in a fixed basis. In many cases, the angles θ are also chosen to implement a specific family of circuits as well [86–88]. Previously, it had not been clear whether this difference is important for the classical complexity or not; our work suggests that it is, as the simulation algorithms we describe crucially rely on the randomness of the circuits for their efficiency and generally require exponential time for the more structured circuits proposed in these works.

B. Our results

We give two classes of results, which we summarize in more detail below. The first consists of rigorous separations in complexity between worst-case simulation [89] and approximate average-case simulation (for sampling) and between near-exact average-case simulation and approximate average-case simulation (for computing output probabilities) for random circuit families defined with respect to certain circuit architectures. While these results are rigorous, they are proved with respect to a contrived architecture and, therefore, do not address the question of whether random shallow circuits are classically simulable more generally. To address this issue, we also give conjectures on the performance of our algorithms for more general and more natural architectures. Our second class of results consists of analytical and numerical evidence supporting these conjectures. A summary of these two

TABLE I.   Summary of results.

Statement	Reference	Proof or evidence
Rigorous complexity separation.—There exists a (nonuniform) shallow 2D random quantum circuit architecture that is efficiently simulable approximately but not exactly in the worst case.	Section IV	Proof for "extended brickwork architecture" using the SEBD algorithm.
Conjecture 1.—There exist uniform shallow 2D random quantum circuit architectures that are efficiently simulable approximately but not exactly in the worst case.	Section III	Reduction from 2D circuit to $1 + 1$D process with alternating layers of unitary gates and weak measurements, which is efficiently simulable when entanglement measures obey an area law. Prior literature indicates area law holds when the weak measurements are sufficiently strong.
	Section V	Numerical implementation of one of our algorithms for two shallow 2D architectures on systems as large as 160 000 qubits; area-law scaling of entanglement entropy suggests asymptotic efficiency.
	Section VI E	Mapping from 2D circuits to Ising-like stat-mech models; whether these models are ordered or disordered determines whether a certain "quasientropy" is in an area- or volume-law phase. We sketch a heuristic argument that sufficiently shallow circuits with sufficiently small local dimension should be in the disordered phase, suggesting algorithmic efficiency.
	Section VI F	Rigorous argument that the stat-mech system for a specific depth-3 uniform 2D architecture is in the disordered phase via direct comparison with the 2D Ising model.
Conjecture $1'$.—When a shallow 2D random quantum circuit architecture is efficiently simulable approximately, its run-time follows a particular dependence on the error, circuit fraction, and number of qubits.	Section III D	Toy model, which predicts a particular entanglement spectrum for states encountered by our algorithm.
	Section V	Numerical observation that entanglement spectrum is consistent with the toy model from Sec. III D and run-time estimate in Conjecture $1'$.
Conjecture 2.—Our algorithms undergo a computational phase transition and become inefficient when either the qudit local dimension or circuit depth exceed some critical value.	Section III	Results from prior literature that similar $1 + 1$D processes undergo entanglement phase transitions.
	Section VI E	Heuristic analysis of stat-mech model for 2D circuits, which predicts order-disorder phase transition and associated entanglement transition for quasientropy.
	Section VI F	Rigorous argument that the stat-mech system for a specific depth-3 uniform 2D architecture undergoes a disorder-order phase transition when the local dimension is increased via direct comparison with the 2D Ising model.

classes of claims along with the evidence we present in their favor appears in Table I.

1. Provable complexity separations

We now summarize our provable results for particular circuit architectures. We first define more precisely what we mean by an "architecture."

Definition 1 (architecture).—An architecture A is an efficiently computable mapping from positive integers L to circuit layouts $A(L)$ defined on rectangular grids with side lengths $L \times f(L)$ for some function $f(L) \leq \text{poly}(L)$. A "circuit layout" is a specification of locations of gates in space and time and the number of qudits acted on by each

gate. (The gate itself is not specified.) For any architecture A, we obtain the associated Haar-random circuit family acting on qudits of constant dimension q, $C_{A,q}$, by specifying every gate in A to be distributed according to the Haar measure and to act on qudits of dimension q which are initialized in a product state $|0\rangle^{\otimes(L \times f(L))}$.

In this paper, we consider only architectures that are constant depth and spatially 2-local (that is, a gate either acts on a single site or two adjacent sites); therefore, "architecture" for our purposes always refers to a constant-depth spatially 2-local architecture. The above definition permits architectures for which the layout of the circuit itself may be different for different sizes. However, it is

natural for a circuit architecture to be spatially periodic and, furthermore, for the "unit cells" of the architecture to be independent of L. We formalize this as a notion of *uniformity*, which we define more precisely below.

Definition 2 (uniformity).—We call a constant-depth architecture A uniform if there exists some spatially periodic circuit layout B on an infinite square lattice such that, for all positive integers L, $A(L)$ is a restriction of B to a rectangular subgrid with side lengths $L \times f(L)$ for some $f(L) \leq \text{poly}(L)$. A random circuit family $C_{A,q}$ associated with a uniform architecture A is said to be a uniform random circuit family.

While uniformity is a natural property for a circuit architecture to possess, our provable separations are with respect to certain nonuniform circuit families. In particular, we prove in Sec. IV that, for any fixed $0 < c < 1$, there exists some nonuniform circuit architecture A acting on n qubits such that, if C_A is the Haar-random circuit family associated with A acting on qubits,

(1) *Exact worst-case sampling is hard.*—There does not exist a $\text{poly}(n)$-time classical algorithm that exactly samples from the output distribution of arbitrary realizations of C_A unless the polynomial hierarchy collapses to the third level.

(2) *Near-exact average-case computation of output probabilities is hard.*—Given an arbitrary fixed output string $\mathbf{x}$, there does not exist a $\text{poly}(n)$-time classical algorithm for computing the probability of obtaining $\mathbf{x}$, $|\langle \mathbf{x} | C_A | 0 \rangle^{\otimes n}|^2$, up to additive error $\leq 2^{-cn \log(n)}$ for a constant $c > 0$, with probability at least $1 - 1/\text{poly}(n)$ over choice of circuit instance, unless a #P-hard function can be computed in randomized polynomial time.

(3) *Approximate average-case sampling is easy.*—There exists a classical algorithm that runs in time $O(n)$ and, with probability at least $1 - 2^{-n^c}$ over choice of circuit instance, samples from the output distribution of C_A up to error at most 2^{-n^c} in total variation distance.

(4) *Approximate average-case computation of output probabilities is easy.*—There exists a classical algorithm that runs in time $O(n)$ and, for an arbitrary output string $\mathbf{x}$, with probability at least $1 - 2^{-n^c}$ over choice of circuit instance, estimates $|\langle \mathbf{x} | C_A | 0 \rangle^{\otimes n}|^2$ up to additive error $2^{-n}/2^{n^c}$. (This should be compared with 2^{-n}, which is the average output probability over choices of $\mathbf{x}$.)

The first two points above follow readily from prior works (respectively, Ref. [5] and Refs. [56,57]), while the latter two follow from an analysis of the behavior of one of our simulation algorithms for this architecture. These algorithms improve on the previously best known simulation time for this family of architectures of $2^{O(L)} = 2^{O(n^{c'})}$ for some constant $c'(c) < 1$ based on an exact simulation based on tensor network contraction. We refer to the

architectures for which we prove the above separations as "extended brickwork architectures" (see Fig. 3 for a specification), as they are related to the "brickwork architecture" [90] studied in the context of MBQC.

Implications for quantum computational supremacy.—The worst-case to average-case reductions that imply the second item above [56,57] are widely cited as evidence for the conjectures that underpin random-circuit-sampling-based quantum computational supremacy proposals [2]. Yet, the items above show that these reductions—which apply directly only to the task of *computing output probabilities*—should not inherently be viewed as evidence for the hardness of the *sampling* tasks of primary interest, as there exist cases in which these reductions apply despite the RCS task being classically tractable. On a more technical level, the existence of an architecture for which both the second and fourth items above are true implies that the most obvious idea for extending these techniques to the realm of sampling problems—namely, by improving the robustness of the interpolations—is impossible. To connect with sampling, the introduction of some new technique that is sensitive to the circuit depth would be required. Thus, although our algorithms can efficiently simulate only shallow random circuits, they accentuate a fundamental weakness in the main source of formal evidence for hardness even in the case of deep circuits. (See Appendix D of Supplemental Material [72] for further discussion of the relationship to this line of work.)

2. Conjectures for uniform architectures

While the above results are provable, they are unfortunately proved with respect to a unnatural nonuniform architecture and, furthermore, do not provide good insight into how the simulation run-time scales with simulation error and simulable circuit fraction. An obvious question is then whether efficient classical simulation remains possible for more natural random circuit families that are sufficiently shallow, and, if so, how the run-time scales with system size and error parameters. We argue that it does but that a computational phase transition occurs for our algorithms when the depth (d) or local Hilbert space dimension (q) becomes too large. Here, we are studying the simulation cost as $n \to \infty$ for fixed d and q. Intuitively, there are many constant-depth random circuit families for which efficient classical simulation is possible, including many "natural" circuit architectures (it seems plausible that *any* depth-3 random circuit family on qubits is efficiently simulable). However, we expect a computational phase transition to occur for sufficiently large constant depths or qudit dimensions, at which point our algorithms become inefficient. The location of the transition point will, in general, depend on the details of the architecture. The conjectures stated below are formalizations of this intuition.

We now state our conjectures more precisely. Conjecture 1 essentially states that there are *uniform* random circuit

families for which worst-case simulation (in the sense of sampling or computing output probabilities) is hard but approximate average-case simulation can be performed efficiently. (Worst-case hardness for computing probabilities also implies a form of average-case hardness for computing probabilities, as discussed above.) This is stated in more or less the weakest form that seems to be true and would yield a polynomial-time simulation. However, we suspect that the scaling is somewhat more favorable. Our numerical simulations and toy models are, in fact, consistent with a stronger conjecture, Conjecture 1′, which if true would yield stronger run-time bounds. Conversely, Conjecture 2 states that if the depth or local qudit dimension of such an architecture is made to be a sufficiently large constant, our two proposed algorithms experience computational phase transitions and become inefficient even for approximate average-case simulation.

Conjecture 1.—There exist uniform architectures and choices of q such that, for the associated random circuit family $C_{A,q}$, (i) worst-case simulation of $C_{A,q}$ (in terms of sampling or computing output probabilities) is classically intractable unless the polynomial hierarchy collapses, and (ii) our algorithms approximately simulate $C_{A,q}$ with high probability. More precisely, given parameters ε and δ, our algorithms run in time bounded by $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$ and can, with probability $1 - \delta$ over the random circuit instance, sample from the classical output distribution produced by C_q up to variational distance error ε and compute a fixed output probability up to additive error ε/q^n.

Conjecture 1′.—For any uniform random circuit family $C_{A,q}$ satisfying the conditions of Conjecture 1, efficient simulation is possible with run-time replaced by

$$n^{1+o(1)} \cdot \exp\{O[\sqrt{\log(1/\varepsilon\delta)}]\}. \qquad (2)$$

Conjecture 2.—For any uniform random circuit family $C_{A,q}$ satisfying the conditions of Conjecture 1, there exists some constant q^* such that our algorithms become inefficient for simulating $C_{A,q'}$ for any constant $q' > q^*$, where $C_{A,q'}$ has the same architecture as C_q but acts on qudits of dimension q'. There also exists some constant k^* such that, for any constant $k > k^*$, our algorithms become inefficient for simulating the composition of k layers of the random circuit, $C_{A,q}^k \circ \cdots \circ C_{A,q}^2 \circ C_{A,q}^1$, where each $C_{A,q}^i$ is independent identically distributed and distributed identically to $C_{A,q}$. In the inefficient regime, for fixed ε and δ the run-time of our algorithms is $2^{O(L)}$.

Our evidence for these conjectures, which we elaborate upon in the following sections, consists primarily of the following elements.

(1) A rigorous reduction from the 2D simulation problem to a 1D simulation problem that can be efficiently solved with high probability if certain conditions on expected entanglement in the 1D state are met (Sec. III).

(2) Convincing numerical evidence that these conditions are indeed met for a specific worst-case-hard uniform random circuit family and that in this case the algorithm is extremely successful in practice (Sec. V).

(3) Heuristic analytical evidence for both conjectures using a mapping from random unitary circuits to classical statistical mechanical models (Sec. VI) and for Conjecture 1′ using a toy model which can be more rigorously studied (Sec. III D).

The uniform random circuit family for which we collect the most evidence for classical simulability is associated with the depth-3 brickwork architecture [90] (see also Fig. 3 for a specification).

In the remainder of the paper, we develop the evidence for our conjectures outlined in the three items above and also present our rigorous complexity separation in Sec. IV.

III. SIMULATION BY REDUCTION TO 1D DYNAMICS

We reduce the problem of simulating a constant-depth quantum circuit acting on a $L \times L'$ grid of qudits to the problem of simulating an associated "effective dynamics" in 1D on L qudits which is iterated for L' time steps or, alternatively, on L' qudits which is iterated for L time steps. This mapping is rigorous and is related to previous maps from 2D quantum systems to 1D system evolving in time [85,91,92]. The effective 1D dynamics is then simulated using the time-evolving block decimation algorithm of Vidal [16]. By analogy, we call this algorithm space-evolving block decimation (SEBD). In Sec. III A, we specify the details of SEBD and rigorously bound the simulation error made by the algorithm in terms of quantities related to the entanglement spectra of the effective 1D dynamics and give conditions in which it is provably asymptotically efficient for sampling and estimating output probabilities with small error. SEBD is self-certifying in the sense that it can construct confidence intervals for its own simulation error and for the fraction of random circuit instances it can simulate. This makes numerically studying the algorithm's performance feasible and is a crucial difference between SEBD and heuristics based on approximate tensor network contractions (e.g., Ref. [70]) in which the error incurred by truncating bonds of the tensor network cannot be directly related to operational measures such as trace-distance error.

A 1D unitary quantum circuit on L qubits iterated for L^c time steps with $c > 0$ is generally hard to simulate classically in $\mathrm{poly}(L)$-time, as the entanglement across any cut can increase linearly in time. However, the form of 1D dynamics that a shallow circuit maps to includes measurements as well as unitary gates. While the unitary gates tend to build entanglement, the measurements tend to destroy entanglement and make classical simulation more tractable. It is *a priori* unclear which effect has more influence.

Fortunately, unitary-and-measurement processes have been studied in a flurry of recent papers from the physics community [22–49]. The consensus from this work is that processes consisting of entanglement-creating unitary evolution interspersed with entanglement-destroying measurements can be in one of two phases, where the entanglement entropy equilibrates to either an area law (constant) or a volume law (extensive). When we vary parameters like the fraction of qudits measured between each round of unitary evolution, a phase transition is observed. The existence of a phase transition appears to be robust to variations in the exact model, such as replacing projective measurements on a fraction of the qudits with weak measurements on all of the qudits [25,26] or replacing Haar-random unitary evolution with Clifford [22,25,27,28] or Floquet [24,25] evolution. This suggests that the efficiency of the SEBD algorithm depends on whether the particular circuit depth and architecture being simulated yields effective 1D dynamics that falls within the area-law or the volume-law regime. It also suggests a computational phase transition in the complexity of the SEBD algorithm. Essentially, decreasing the measurement strength or increasing the qudit dimension in these models is associated with moving toward a transition into the volume-law phase. Since increasing the 2D circuit depth is associated with decreasing the measurement strength and increasing the local dimension of the associated effective 1D dynamics, this already gives substantial evidence in favor of a computational phase transition in SEBD.

SEBD is inefficient if the effective 1D dynamics are on the volume-law side of the transition, and we expect it to be efficient on the area-law side, because, in practice, dynamics obeying an area law for the von Neumann entanglement entropy are generally efficiently simulable. However, definitely proving that SEBD is efficient on the area-law side faces the obstacle that there are known contrived examples of states which obey an area law but cannot be efficiently simulated with matrix product states (MPSs) [93]. We address this concern by directly studying the entanglement spectrum of unitary-and-measurement processes in the area-law phase. To do this, we introduce a toy model for such dynamics which may be of independent interest. For this model, discussed more in Sec. III D, we rigorously derive an asymptotic scaling of Schmidt values across some cut as $\lambda_i \propto \exp[-\Theta(\log^2 i)]$ which is consistent with the scaling observed in our numerical simulations. Moreover, for this toy model, we show that with probability at least $1 - \delta$, the equilibrium state after iterating the process can be ε-approximated by a state with Schmidt rank $r \leq \exp\{O[\sqrt{\log(n/\varepsilon\delta)}]\}$. Taking this toy model analysis as evidence that the bond dimension of SEBD when simulating a circuit whose effective 1D dynamics is in an area-law phase obeys this asymptotic scaling leads to Conjecture 1′.

A. Specification of algorithm

In this section, we assume the reader is familiar with standard tensor network methods, particularly algorithms for manipulating matrix product states (see, e.g., Refs. [94,95] for reviews).

For concreteness, we consider a rectangular grid of qudits with local Hilbert space dimension q, although the algorithm could be similarly defined for different lattices. Assume without loss of generality that the grid consists of $n = L_1 \times L_2$ qudits, where L_1 is the number of rows, L_2 is the number of columns, and $L_1 \leq L_2$. For each qudit, let $|i\rangle, i \in [q] := \{0, 1, ..., q - 1\}$, label a set of basis states which together form the computational basis. Assume all gates act on one site or two neighboring sites, and the starting state is $|0\rangle^{\otimes n}$. Let d denote the circuit depth, which should be regarded as a constant. For a fixed circuit instance C, the goal is to sample from a distribution close to $\mathcal{D}_C$, defined to be the distribution of the output of C upon measuring all qudits in the computational basis. For an output string $\mathbf{x} \in [q]^n$, we let $\mathcal{D}_C(\mathbf{x})$ denote the probability of the circuit outputting $\mathbf{x}$ after measurement. The high-level behavior of the algorithm is illustrated in Fig. 1. Recall that C can always be *exactly* simulated in time $L_2 q^{\Theta(dL_1)}$ using standard tensor network algorithms [13].

Since all of the single-qudit measurements commute, we can measure the qudits in any order. In particular, we can
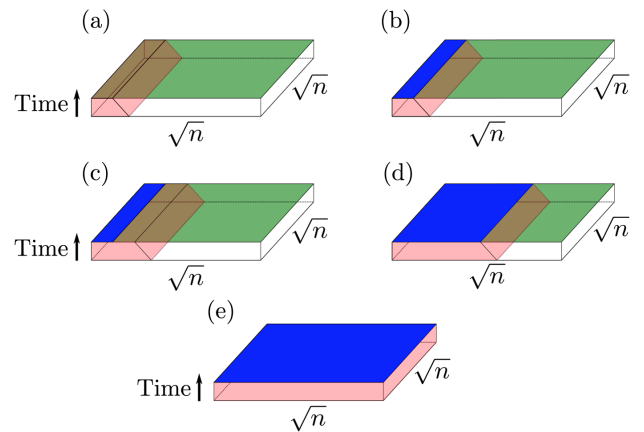

FIG. 1.   Schematic depiction of SEBD simulating a shallow 2D circuit. In all figures, the 2D circuit is depicted as a spacetime volume, with time flowing upward. The blue regions correspond to sites for which measurements have been simulated, while green regions correspond to unmeasured sites. In (a), we apply all gates in the light cone of column 1, namely, those gates intersecting the spacetime volume shaded red. In (b), we simulate the computational basis measurement of column 1. In (c), we apply all gates in the light cone of column 2 that were previously unperformed. (d) depicts the algorithm at an intermediate stage of the simulation, after the measurements of about half of the qudits have been simulated. The algorithm stores the current state as an MPS at all times, which may be periodically compressed to improve efficiency. (e) depicts the algorithm at completion: The measurements of all n of the qudits are simulated.

first measure all of the sites in column 1, then those in column 2, and iterate until we have measured all L_2 columns. This is the measurement order we take. Now, consider the first step in which we measure column 1. Instead of applying all of the gates of the circuit and then measuring, we may instead apply only the gates in the *light cone* of column 1, that is, the gates that are causally connected to the measurements in column 1. We may ignore qudits that are outside the light cone, by which we mean qudits that are outside the support of all gates in the light cone.

Let $\rho_1 = |0\rangle\langle 0|^{\otimes L_1}$ denote the trivial starting state that is a tensor product of $|0\rangle$ states in column 1, which the algorithm represents as an MPS. Let V_1 denote the isometry corresponding to applying all gates in the light cone of this column. The algorithm simulates the application of V_1 by adding qudits in the light cone of column 1 as necessary and applying the associated unitary gates, maintaining the description of the state as an MPS of length L_1 as illustrated in Fig. 2. Since there are up to $d + 1$ columns in the light cone of column 1, each tensor of the MPS after the application of V_1 has up to $d + 1$ dangling legs corresponding to physical indices, for a total physical dimension of at most q^{d+1}. Since in the application of V_1 there are up to $O(d^2)$ gates that act between any two neighboring rows,
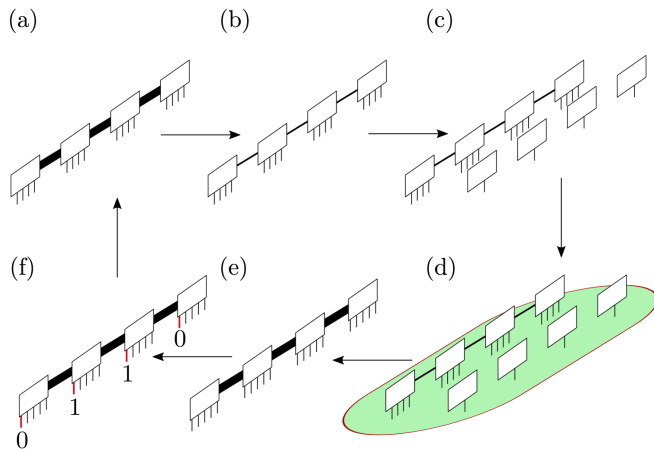

(a)   (b)   (c)

(f)   (e)   (d)

FIG. 2. Iteration of SEBD. In (a), we begin with an MPS describing the current state ρ_j. In (b), the MPS is compressed via truncation of small Schmidt values. This generally decreases the bond dimension of the MPS, depicted in the cartoon by a reduction in the thickness of the lines between tensors. In (c), qudits acted on by V_j that are not already incorporated into the current state are added to the MPS (increasing the physical bond dimension of the MPS) and initialized in $|0\rangle$ states. In (d), the unitary gates associated with V_j are applied. (e) depicts the MPS after the application of V_j; the virtual bond dimension generally is increased by the application of V_j. In (f), the measurement of column j is performed, and the outcome 0110 is obtained. Subsequently, column j is projected onto the outcome 0110, removing the physical legs associated with these sites from the MPS. The resulting state is ρ_{j+1}.

the (virtual) bond dimension of the updated MPS is at most $q^{O(d^2)}$.

We now simulate the computational basis measurement of column 1. More precisely, we measure the qudits of column 1 one by one. We first compute the respective probabilities $p_1, p_2, \ldots, p_q$ of the q possible measurement outcomes for the first qudit. This involves contracting the MPS encoding $V_1\rho_1 V_1^\dagger$. We now use these probabilities to classically sample an outcome $i \in [q]$ and update the MPS to condition on this outcome. That is, if (say) we obtain outcome 1 for site i, we apply the projector $|0\rangle\langle 0|$ to site i of the state and subsequently renormalize. After doing this for every qudit in the column, we have exactly sampled an output string $\mathbf{x}_1 \in [q]^{L_1}$ from the marginal distribution on column 1 and are left with an MPS description of the pure, normalized, postmeasurement state ρ_2 proportional to $\mathrm{tr}_{\text{column 1}}(\Pi_1^{\mathbf{x}} V_1\rho_1 V_1^\dagger \Pi_1^{\mathbf{x}})$, where $\Pi_1^{\mathbf{x}}$ denotes the projection of column 1 onto the sampled output string $\mathbf{x} = \mathbf{x}_1$. Using standard tensor network algorithms, the time complexity of these steps is $L_1 q^{O(d^2)}$.

We next consider column 2. At this point, we add the qudits and apply the gates that are in the light cone of column 2 but were not applied previously. Denote this isometry by V_2. It is straightforward to see that this step respects causality. That is, if some gate U is in the light cone of column 1, then any gate W that is in the light cone of column 2 but not column 1 cannot be required to be applied before U, because if it were, then it would be in the light cone of column 1. Hence, when we apply gates in this step, we never apply a gate that is required to be applied before some gate that is applied in the first step. After this step, we apply all gates in the light cone of columns 1 and 2, and we also project column 1 onto the measurement outcomes we observe.

By simulating the measurements of column 2 in a similar way to those of column 1, we sample a string $\mathbf{x}_2$ from the marginal distribution on column 2, conditioned on the previously observed outcomes from column 1. Each time an isometry V_j is applied, the bond dimension of the MPS representation of the current state, in general, increases by a multiplicative factor. In particular, if we iterate this procedure to simulate the entire lattice, we eventually encounter a maximal bond dimension of up to $q^{O(dL_1)}$ and obtain a sample $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{L_2}) \in [q]^n$ from the true output distribution.

To improve the efficiency at the expense of accuracy, we may compress the MPS in each iteration to one with smaller bond dimension using standard MPS compression algorithms. In particular, in each iteration j before we apply the corresponding isometry V_j, we first discard as many of the smallest singular values (i.e., Schmidt values) associated with each cut of the MPS as possible up to a total *truncation error* per bond of ϵ, defined as the sum of the squares of the discarded singular values. The bond dimension across any

cut is reduced by the number of discarded values. This truncation introduces some error that we quantify below.

If the maximal bond dimension of this truncated version of the simulation algorithm is D, the total run-time of the full algorithm to obtain a sample is bounded by (taking q and d to be constants) $O(nD^3)$ using standard MPS compression algorithms.

We assume that for a specified maximal bond dimension D and truncation error per bond ϵ, if a bond dimension ever exceeds D, then the algorithm terminates and outputs a failure flag fail. Hence, the run-time of the algorithm when simulating some circuit C with parameters ϵ and D is bounded by $O(nD^3)$, and the algorithm has some probability of failure $p_{f,C}$. We summarize the SEBD algorithm in Algorithm 1.

The untruncated version of the algorithm presented above samples from the true distribution $\mathcal{D}_C$ of the measurement outcomes of the original 2D circuit C. However, due to the MPS compression which we perform in each iteration and the possibility of failure, the algorithm incurs some error which causes it to instead sample from some distribution $\mathcal{D}'_C$. Here, we bound the total variation distance between these distributions, given by

$$\frac{1}{2}\|\mathcal{D}'_C - \mathcal{D}_C\|_1 = \frac{1}{2}\sum_{\mathbf{x}}|\mathcal{D}'_C(\mathbf{x}) - \mathcal{D}_C(\mathbf{x})| + \frac{1}{2}p_{f,C}, \quad (3)$$

where the sum runs over the q^n possible output strings (not including fail), in terms of the truncation error made by the algorithm.

Algorithm 1. SEBD

Input: circuit instance C, truncation error ϵ, bond dimension cutoff D

Output: string $\mathbf{x} \in [q]^n$ or fail

Run-time: $O(nD^3)$ [q and d assumed to be constants]

1: initialize an MPS in the state $|0\rangle\langle 0|^{\otimes L_1}$, corresponding to column 1

2: **for** $t = 1\dots L_2$ **do**

3:      compress MPS describing state by truncating small singular values, up to error ϵ per bond

4:      apply V_t, corresponding to gates in the light cone of not yet applied

5:      if some bond dimension is greater than D, terminate and output fail

6:      simulate measurement of all qudits in column t via MPS contraction and sampling

7:      apply $\Pi_t^{\mathbf{x}_t}$ to condition on measurement string $\mathbf{x}_t$ observed for that column

return $(\mathbf{x}_1, \dots, \mathbf{x}_{L_2}) \in [q]^n$

We first obtain a very general bound on the error made by SEBD with no bond dimension cutoff in terms of the truncation error. Note that the truncation error may depend on the (random) measurement outcomes and is itself, therefore, a random variable. See Appendix E in Supplemental Material [72] for a proof.

Lemma 1.—Let ϵ_i denote the sum of the squares of all singular values discarded in the compression during iteration i of the simulation of a circuit C with output distribution $\mathcal{D}_C$ by SEBD with no bond dimension cutoff, and let Λ denote the sum of all singular values discarded over the course of the algorithm. Then, the distribution $\mathcal{D}'_C$ sampled from by SEBD satisfies

$$\frac{1}{2}\|\mathcal{D}'_C - \mathcal{D}_C\|_1 \le \mathbb{E}\sum_{i=1}^{L_2}\sqrt{2\epsilon_i} \le \sqrt{2}\mathbb{E}\Lambda, \quad (4)$$

where the expectations are over the random measurement outcomes.

From Lemma 1, we immediately obtain two corollaries. The first is useful for empirically bounding the sampling error in total variation distance made by SEBD when the algorithm also has a bond dimension cutoff. The second is a useful asymptotic statement. The corollaries follow straightforwardly from the coupling formulation of variational distance, Markov's inequality, and the triangle inequality.

Corollary 1.—Let $\mathcal{A}$ denote a SEBD algorithm with truncation error parameter ϵ and bond dimension cutoff D. Consider a fixed circuit C, and suppose that $\mathcal{A}$ applied to this circuit fails with probability $p_{f,C}$. Then $\mathcal{A}$ samples from the output distribution of C with total variation distance error bounded by $L_2\sqrt{2\epsilon L_1} + p_{f,C}$.

If the failure probability of $\mathcal{A}$ averaged over random choice of circuit instance and measurement outcome is p_f, then for any δ, on at least $1 - \delta$ fraction of circuit instances, $\mathcal{A}$ samples from the true output distribution with a total variation distance error bounded by $L_2\sqrt{2\epsilon L_1} + p_f/\delta$.

In practice, the variational distance error of SEBD with truncation error ϵ applied to the simulation of some circuit C can be bounded by constructing a confidence interval for $p_{f,C}$ and applying the above bound.

Corollary 2.—Let $\mathcal{A}$ denote a SEBD algorithm with truncation error parameter ϵ and no bond dimension cutoff. Suppose that, for some random circuit family with $q = O(1)$ and $d = O(1)$, the expected bond dimension across any cut is bounded by $\text{poly}(n, 1/\epsilon)$. Then, SEBD with some choice of $\epsilon = 1/\text{poly}(n)$ and $D = \text{poly}(n)$ runs in time $\text{poly}(n, 1/\varepsilon, 1/\delta)$ and, with probability at least $1 - \delta$ over the choice of circuit instance C, samples from the output distribution of C with variational distance error less than ε.

Thus, to prove the part of Conjecture 1 about sampling up to total variation distance error ε for uniform random circuit families, it would suffice to show that there is a 2D

constant-depth uniform random quantum circuit family with the worst-case-hard property for which the expected bond dimension across any cut while running SEBD with truncation parameter ϵ is bounded by $\mathrm{poly}(n, 1/\epsilon)$. Later, we introduce two candidate circuit families for which we can give numerical and analytical evidence that this criterion is indeed met.

In the next subsection, we show how the other part of Conjecture 1, regarding computing output probabilities, also follows from a $\mathrm{poly}(n, 1/\epsilon)$ bound on the bond dimension of states encountered by SEBD on uniform worst-case-hard circuit families.

B. Computing output probabilities with SEBD

In the previous section, we describe how a SEBD algorithm with a truncation error parameter ϵ and a bond dimension cutoff D applied to a circuit C samples from a distribution $\mathcal{D}'_C$ satisfying $\|\mathcal{D}'_C - \mathcal{D}_C\|_1 \leq 2L_2\sqrt{2\epsilon L_1} + 2p_{f,C}$, where $p_{f,C}$ is the probability that some bond dimension exceeds D and the algorithm terminates and indicates failure. Expanding the expression for the 1-norm and rearranging, we have

$$\frac{1}{q^n}\sum_{\mathbf{x}}|\mathcal{D}'_C(\mathbf{x}) - \mathcal{D}_C(\mathbf{x})| \leq \frac{2L_2\sqrt{2\epsilon L_1} + p_{f,C}}{q^n}. \quad (5)$$

SEBD with bond dimension cutoff D can be used to compute $\mathcal{D}'_C(\mathbf{x})$ for any output string $\mathbf{x}$ in time $O(nD^3)$ (taking q and d to be constants). To do this, for a fixed output string $\mathbf{x}$, SEBD proceeds similarly to the case in which it is being used for sampling, but, rather than sampling from the output distribution of some column, it simply projects that column onto the outcome specified by the string $\mathbf{x}$ and computes the conditional probability of that outcome via contraction of the MPS. That is, at iteration t, the algorithm computes the conditional probability of measuring the string $\mathbf{x}_t \in [q]^{L_1}$ in column t, $\mathcal{D}'_C(\mathbf{x}_t|\mathbf{x}_1, ..., \mathbf{x}_{t-1})$, by projecting column t onto the relevant string via the projector $\Pi_t^{\mathbf{x}_t}$ and then contracting the relevant MPS. If the bond dimension ever exceeds D, then it must hold that $\mathcal{D}'_C(\mathbf{x}) = 0$, and so the algorithm outputs zero and terminates. Otherwise, the algorithm outputs $\mathcal{D}'_C(\mathbf{x}) = \prod_{t=1}^{L_2}\mathcal{D}'_C(\mathbf{x}_t|\mathbf{x}_1, ..., \mathbf{x}_{t-1})$. We summarize this procedure in Algorithm 2.

We have therefore shown the following.

Lemma 2.—Let $p_{f,C}$ be the failure probability of SEBD when used to simulate a circuit instance C with truncation error parameter ϵ and bond dimension cutoff D. Suppose $\mathbf{x} \in [q]^n$ is an output string drawn uniformly at random. Then Algorithm 2 outputs a number $\mathcal{D}'_C(\mathbf{x})$ satisfying

$$\mathbb{E}_{\mathbf{x}}|\mathcal{D}'_C(\mathbf{x}) - \mathcal{D}_C(\mathbf{x})| \leq \frac{2L_2\sqrt{2\epsilon L_1} + p_{f,C}}{q^n}. \quad (6)$$

Algorithm 2. SEBD for computing output probabilities.

Input: circuit instance C, truncation error ϵ, bond dimension cutoff D, string $\mathbf{x} \in [q]^n$

Output: $\mathcal{D}'_C(\mathbf{x})$

Run-time: $O(nD^3)$ [q and d assumed to be constants]

1: initialize an MPS in the state $|0\rangle\langle 0|^{\otimes L_1}$, corresponding to column 1

2: **for** $t = 1...L_2$ **do**

3:　　compress MPS describing state by truncating small singular values, up to error ϵ per bond

4:　　apply V_t, corresponding to gates in the light cone of column t not yet applied

5:　　if some bond dimension is greater than D, terminate and output zero

6:　　apply $\Pi_t^{\mathbf{x}_t}$ to condition on string $\mathbf{x}_t$

7:　　compute $\mathcal{D}'_C(\mathbf{x}_t|\mathbf{x}_1, ..., \mathbf{x}_{t-1})$ via MPS contraction

return $\mathcal{D}'_C(\mathbf{x}) = \prod_{t=1}^{L_2}\mathcal{D}'_C(\mathbf{x}_t|\mathbf{x}_1, ..., \mathbf{x}_{t-1})$

The above lemma bounds the expected error incurred while estimating a uniformly random output probability for a fixed circuit instance C. We may use this lemma to straightforwardly bound the expected error incurred while estimating the probability of a fixed output string over a distribution of random circuit instances. The corollary is applicable if the distribution of circuit instances has the property of being invariant under an application of a final layer of arbitrary single-qudit gates. This includes circuits in which all gates are Haar-random (as long as every qudit is acted on by some gate) but is more general. In particular, any circuit distribution in which the final gate to act on any given qudit is Haar-random satisfies this property. This fact is relevant in subsequent sections.

Corollary 3.—Let p_f be the failure probability of SEBD when used to simulate a random circuit instance C with truncation error parameter ϵ and bond dimension cutoff D, where C is drawn from a distribution that is invariant under application of a final layer of arbitrary single-qudit gates. Then for any fixed string $\mathbf{x} \in [q]^n$ the output of Algorithm 2 satisfies

$$\mathbb{E}_C|\mathcal{D}'_C(\mathbf{x}) - \mathcal{D}_C(\mathbf{x})| \leq \frac{2L_2\sqrt{2\epsilon L_1} + p_f}{q^n}. \quad (7)$$

Proof.—Averaging the bound of Eq. (6) over random circuit instances, we have

$$\mathbb{E}_{\mathbf{y}}\mathbb{E}_C|\mathcal{D}'_C(\mathbf{y}) - \mathcal{D}_C(\mathbf{y})| \leq \frac{2L_2\sqrt{2\epsilon L_1} + p_f}{q^n}. \quad (8)$$

Let $L_{\mathbf{y}}$ denote a layer of single-qudit gates with the property that $L_{\mathbf{y}}|\mathbf{x}\rangle = |\mathbf{y}\rangle$. By assumption, C is distributed

identically to the composition of C with $L_\mathbf{y}$, denoted $L_\mathbf{y} \circ C$. Together with the observation that $\mathcal{D}_{L_\mathbf{y} \circ C}(\mathbf{y}) = \mathcal{D}_C(\mathbf{x})$, we have

$$\mathbb{E}_\mathbf{y} \mathbb{E}_C |\mathcal{D}'_C(\mathbf{y}) - \mathcal{D}_C(\mathbf{y})| = \mathbb{E}_\mathbf{y} \mathbb{E}_C |\mathcal{D}'_{L_\mathbf{y} \circ C}(\mathbf{y}) - \mathcal{D}_{L_\mathbf{y} \circ C}(\mathbf{y})| \quad (9)$$

$$= \mathbb{E}_C |\mathcal{D}'_C(\mathbf{x}) - \mathcal{D}_C(\mathbf{x})|, \quad (10)$$

from which the result follows. ∎

The following asymptotic statement follows straightforwardly.

Corollary 4.—Let $\mathcal{A}$ denote a SEBD algorithm with truncation error parameter ϵ and no bond dimension cutoff. Suppose that, for some random circuit family with $q = O(1)$ and $d = O(1)$, the expected bond dimension across any cut is bounded by $\mathrm{poly}(n, 1/\epsilon)$. Then, SEBD with some choice of $\epsilon = 1/\mathrm{poly}(n)$ and $D = \mathrm{poly}(n)$ runs in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ and, with probability at least $1 - \delta$ over the choice of circuit instance C, estimates $\mathcal{D}_C(\mathbf{x})$ for some fixed $\mathbf{x} \in [q]^n$ up to additive error bounded by ϵ/q^n.

Corollary 4 shows how the part of Conjecture 1 about computing arbitrary output probabilities to error ϵ/q^n would follow from a bound on the bond dimension across any cut when SEBD runs on a uniform worst-case-hard circuit family.

C. Example: SEBD applied to cluster state with Haar-random measurements (CHR)

To illustrate the connection between SEBD and random-unitary-and-measurement dynamics, we now study the SEBD algorithm in more detail for a simple uniform family of 2D random circuits that possesses the worst-case-hard property required by Conjecture 1. The model we consider is the following: Start with a 2D cluster state of n qubits arranged in a $\sqrt{n} \times \sqrt{n}$ grid, apply a single-qubit Haar-random gate to each qubit, and then measure all qubits in the computational basis. Recall that a cluster state may be created by starting with the product state $|+\rangle^{\otimes n}$ before applying CZ gates between all adjacent sites. An equivalent formulation which we find convenient in the subsequent section is to measure each qubit of the cluster state in a Haar-random basis. We refer to this model as CHR, for "cluster state with Haar-random measurements."

Following Ref. [8], it is straightforward to show that sampling from the output distribution of CHR is classically *worst-case hard* assuming the polynomial hierarchy (PH) does not collapse to the third level. It can also be readily shown, following Ref. [54], that near-exactly computing output probabilities of CHR is #P-hard in the average case. These results rule out, under standard conjectures, the existence of a classical sampling algorithm for CHR that succeeds for all instances or a classical algorithm for efficiently computing most output probabilities of CHR near exactly. A natural question is then whether efficient

approximate average-case versions of these algorithms may exist. We formalize these questions as the problems $\mathrm{CHR}^{\mathrm{samp/prob}}_\pm$.

Problem 1 ($\mathrm{CHR}^{\mathrm{samp/prob}}_\pm$).—Given as input a random instance C of CHR (specified by a side length $\sqrt{n}$ and a set of n single-qubit Haar-random gates applied to the $\sqrt{n} \times \sqrt{n}$ cluster state) and error parameters ϵ and δ, perform the following computational task in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

(i) $\mathrm{CHR}^{\mathrm{samp}}_\pm$.—Sample from a distribution $\mathcal{D}'_C$ that is ϵ-close in total variation distance to the true output distribution $\mathcal{D}_C$ of circuit C, with probability of success at least $1 - \delta$ over the choice of measurement bases.

(ii) $\mathrm{CHR}^{\mathrm{prob}}_\pm$.—Estimate $\mathcal{D}_C(\mathbf{0})$, the probability of obtaining the all-zeros string upon measuring the output state of C in the computational basis, up to additive error at most $\epsilon/2^n$, with probability of success at least $1 - \delta$ over the choice of measurement bases.

We now show that SEBD solves $\mathrm{CHR}^{\mathrm{samp/prob}}_\pm$ if a certain form of 1D dynamics involving local unitary gates and measurements is classically simulable. We first consider the sampling variant of SEBD. Specializing to the CHR model, the algorithm takes on a particularly simple form due to the fact that the cluster state is built by applying CZ gates between all neighboring pairs of qubits, which are initialized in $|+\rangle$ states. Because of this structure, the radius of the light cone for this model is simply one. In particular, the only gates in the light cone of columns 1–j are the Haar-random single-qubit gates acting on qubits in these columns, as well as CZ gates that act on at least one qubit within these columns. This permits a simple prescription for SEBD applied to this problem.

Initialize the simulation algorithm in the state $\rho_1 = |+\rangle\langle+|^{\otimes\sqrt{n}}$ corresponding to column 1. To implement the isometry V_1, initialize the qubits of column 2 in the state $|+\rangle\langle+|^{\otimes\sqrt{n}}$ and apply CZ gates between adjacent qubits that are both in column 1 and between adjacent qubits in separate columns. Now, measure the qubits of column 1 in the specified Haar-random bases (equivalently, apply the specified Haar-random gates and measure in the computational basis), inducing a pure state ρ_2 with support in column 2. Iterating this process, we progress through a random sequence of 1D states on $\sqrt{n}$ qubits $\rho_1 \to \rho_2 \to \cdots \to \rho_{\sqrt{n}}$, which we see can be equivalently understood as arising from a 1D dynamical process consisting of alternating layers of random unitary gates and weak measurements.

It is helpful to introduce notation. Define $|\theta, \phi\rangle := \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$. In other words, let $|\theta, \phi\rangle$ denote the single-qubit pure state with polar angle θ and azimuthal angle ϕ on the Bloch sphere. Let $\theta_i^{(t)}$ and $\phi_i^{(t)}$ specify the measurement basis of the qubit in row i and column t; that is, the projective measurement on the qubit in

row i and column t is $\{\Pi^0_{\theta^{(t)}_i,\phi^{(t)}_i}, \Pi^1_{\theta^{(t)}_i,\phi^{(t)}_i}\}$ with $\Pi^0_{\theta^{(t)}_i,\phi^{(t)}_i} :=$ $|\theta^{(t)}_i,\phi^{(t)}_i\rangle\langle\theta^{(t)}_i,\phi^{(t)}_i|$ and $\Pi^1_{\theta^{(t)}_i,\phi^{(t)}_i} := I - \Pi^0_{\theta^{(t)}_i,\phi^{(t)}_i}$. We also define

$$M_0(\theta,\phi) := \begin{pmatrix} \cos(\theta/2) & 0 \\ 0 & e^{-i\phi}\sin(\theta/2) \end{pmatrix}, \qquad (11\text{a})$$

$$M_1(\theta,\phi) := \begin{pmatrix} \sin(\theta/2) & 0 \\ 0 & e^{i\phi}\cos(\theta/2) \end{pmatrix}. \qquad (11\text{b})$$

Note that $\{M_0(\theta,\phi), M_1(\theta,\phi)\}$ defines a weak single-qubit measurement. We now describe, in Algorithm 3, a 1D process which we claim produces a sequence of states identical to that encountered by SEBD for the same choice of measurement bases and measurement outcomes and also has the same measurement statistics.

Lemma 3.—For a fixed choice of $\{\theta^{(t)}_i, \phi^{(t)}_i\}$ parameters, the joint distribution of outcomes $\{X^{(t)}_i\}_{i,t}$ is identical to that of $\{Y^{(t)}_i\}_{i,t}$, where $\{Y^{(t)}_i\}_{i,t}$ are the measurement outcomes obtained upon measuring all qubits of a $\sqrt{n} \times \sqrt{n}$ cluster state, with the measurement on the qubit in row i and column t being $\{\Pi^0_{\theta^{(t)}_i,\phi^{(t)}_i}, \Pi^1_{\theta^{(t)}_i,\phi^{(t)}_i}\}$. Furthermore, for any fixed choice of measurement outcomes, $\varphi_j = \rho_j$ for all $j \in \{1,\dots,\sqrt{n}\}$, where ρ_j is the state at the beginning of iteration j of the SEBD algorithm.

Proof.—The lemma follows from the above description of the behavior of SEBD applied to CHR, as well as the following identities holding for any single-qubit state $|\xi\rangle$ which may be verified by straightforward calculation:

$$(\Pi^0_{\theta,\phi} \otimes I)CZ(|\xi\rangle \otimes |+\rangle) = |\theta,\phi\rangle \otimes HM_0(\theta,\phi)|\xi\rangle, \qquad (12)$$

Algorithm 3. Effective 1D dynamics of a fixed instance of CHR.

1: $\varphi_1 \leftarrow |+\rangle\langle+|^{\otimes\sqrt{n}}$.

2: **for** $t = 1\dots\sqrt{n}-1$ **do**

3:     apply a CZ gate between every adjacent pair of qubits

4:     measure $\{M_0(\theta^{(t)}_i,\phi^{(t)}_i), M_1(\theta^{(t)}_i,\phi^{(t)}_i)\}$ on qubit i,   obtaining $X^{(t)}_i$, for $i \in \{1,\dots,\sqrt{n}\}$

5:     apply a Hadamard transform

6:     $\varphi_{t+1} \leftarrow$ resulting state

7: measure $\{\Pi^0_{\theta^{(\sqrt{n})}_i,\phi^{(\sqrt{n})}_i}, \Pi^1_{\theta^{(\sqrt{n})}_i,\phi^{(\sqrt{n})}_i}\}$ on qubit i, obtaining $X^{(\sqrt{n})}_i$, for $i \in \{1,\dots,\sqrt{n}\}$

$$(\Pi^1_{\theta,\phi} \otimes I)CZ(|\xi\rangle \otimes |+\rangle) = |\pi-\theta,-\phi\rangle \otimes HM_1(\theta,\phi)|\xi\rangle. \qquad (13)$$

∎

We see that, for a fixed choice of single-qubit measurement bases $\{\theta^{(t)}_j, \phi^{(t)}_j\}_{t,j}$ associated with an instance C, we can define an associated 1D process consisting of alternating layers of single-qubit weak measurements and local unitary gates, such that simulating this 1D process is sufficient for sampling from $\mathcal{D}_C$.

Now, recall that, in the context of simulating CHR, each single-qubit measurement basis is chosen randomly according to the Haar measure. That is, the Bloch sphere angles $(\theta^{(t)}_i, \phi^{(t)}_i)$ are Haar-distributed. If we define $x^{(t)}_i \equiv \cos\theta^{(t)}_i$, we find that $x^{(t)}_i$ is uniformly distributed on the interval $[-1,1]$. The parameters $\phi^{(t)}_i$ are uniformly distributed on $[0, 2\pi]$. Using these observations, as well as the observation that the outcome probabilities of the measurement of qubit i in iteration t are independent of the azimuthal angle $\phi^{(t)}_i$ when $t < \sqrt{n}$, we may derive effective dynamics of a random instance.

Define the operators

$$N(x) := \begin{pmatrix} \sqrt{\frac{1+x}{2}} & 0 \\ 0 & \sqrt{\frac{1-x}{2}} \end{pmatrix}, \qquad x \in [-1,1].$$

Note that $\{N(x), N(-x)\}$ defines a weak measurement. Also, define the phase gate

$$P(\phi) := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}, \qquad \phi \in [0, 2\pi].$$

By randomizing each single-qubit measurement basis according to the Haar distribution, one finds that the dynamics of Algorithm 3 (which applies for a fixed choice of measurement bases) may be written as Algorithm 4 below, where the notation $x \in_U [-1, 1]$ means that x is a random variable uniformly distributed on $[-1, 1]$. That is, the distribution of random sequences $\varphi_1 \to \varphi_2 \to \cdots \to \varphi_{\sqrt{n}}$ and distribution of output statistics produced by Algorithm 4 is identical to that produced by SEBD applied to CHR.

Hence, if time-evolving block decimation (TEBD) can efficiently simulate the process of Algorithm 4 with high probability, then SEBD can solve $\mathrm{CHR}^{\mathrm{samp}}_{\pm}$ and $\mathrm{CHR}^{\mathrm{prob}}_{\pm}$. We formalize this in the following lemma.

Lemma 4.—Suppose that TEBD can efficiently simulate the process described in Algorithm 4 in the sense that the expected bond dimension across any cut is bounded by

---

Algorithm 4. Effective 1D dynamics of CHR.

1: $\varphi_1 \leftarrow |+\rangle\langle+|^{\otimes\sqrt{n}}$.

2: **for** $t = 1 \ldots \sqrt{n} - 1$ **do**

3:     apply a CZ gate between every adjacent pair of qubits

4:     **for** $i = 1 \ldots \sqrt{n}$ **do**

5:         measure $\{N(x), N(-x)\}$ on qubit i with $x \in_U [-1, 1]$

6:         apply the gate $P(\phi)$ with $\phi \in_U [0, 2\pi]$ to qubit i

7:     apply a Hadamard transform

8:     $\varphi_{t+1} \leftarrow$ resulting state

9: perform a projective measurement on each qubit in a
   Haar-random basis

$\text{poly}(n, 1/\epsilon)$, where ϵ is the truncation error parameter. Then, SEBD can be used to solve $\text{CHR}_\pm^{\text{samp}}$ and $\text{CHR}_\pm^{\text{prob}}$.

Proof.—The proof follows from Corollary 2, Corollary 4, and the equivalence to Algorithm 4 discussed above. ∎

We have shown how SEBD applied to CHR can be reinterpreted as TEBD applied to a 1D dynamical process involving alternating layers of random unitaries and weak measurements. Up until this point, there has been little reason to expect that SEBD is efficient for the simulation of CHR. In particular, with no truncation, the bond dimension of the MPS stored by the algorithm grows exponentially as the algorithm sweeps across the lattice.

We now invoke the findings of a number of related recent works [22–49] to motivate the possibility that TEBD can efficiently simulate the effective 1D dynamics. These works study various 1D dynamical processes involving alternating layers of measurements and random local unitaries. In some cases, the measurements are considered to be projective and occur only with some probability p. In other cases, similarly to Algorithm 4, weak measurements are applied to each site with probability one. The common finding of these papers is that such models appear to exhibit an entanglement phase transition driven by measurement probability p (in the former case) or measurement strength (in the latter case). On one side of the transition, the entanglement entropy obeys an area law, scaling as $O(1)$ with the length L. On the other side, it obeys a volume law, scaling as $O(L)$.

Based on these works, one expects the entanglement dynamics to saturate to an area-law or volume-law phase. And in fact, our numerical studies (presented in Sec. V) suggest that these dynamics saturate to an area-law phase. The common intuition that 1D quantum systems obeying an area law for the von Neumann entropy are easy to simulate with matrix product states therefore suggests that SEBD applied to this problem is efficient. While counterexamples to this common intuition are known [93], they are contrived and do not present an obvious obstruction for our

algorithm. To better understand the relationship between maximal bond dimension and truncation error when the effective dynamics is in the area-law phase as well as rule out such counterexamples, in the following section, we describe a toy model for a unitary-and-measurement process in the area-law phase, which predicts a super-polynomial decay of Schmidt values across any cut and, therefore, predicts that a polynomial run-time is sufficient to perform the simulation to $1/\text{poly}(n)$ error. Our numerical results (presented in Sec. V) suggest that the effective dynamics of the random circuit architectures we consider are indeed in the area-law phase, with entanglement spectra consistent with those predicted by the toy model dynamics. Further analytical evidence for efficiency is given in Sec. VI.

Note that, although we explicitly derive the effective 1D dynamics for the CHR model and observe it to be a simple unitary-and-measurement process, the interpretation of the effective 1D dynamics as a unitary-and-measurement process is not specific to CHR and is, in fact, general. In the general case, SEBD tracks $O(r)$ columns simultaneously where r is the radius of the light cone corresponding to the circuit. In each iteration, new qudits that have come into the light cone are added, unitary gates that have come into the light cone are performed, and finally projective measurements are performed on a single column of qudits. Similarly to the case of CHR, this entire procedure can be viewed as an application of unitary gates followed by weak measurements on a 1D chain of qudits of dimension $q^{O(r)}$. Intuitively, increasing the circuit depth corresponds to both increasing the local dimension in the effective 1D dynamics and decreasing the measurement strength. The former is due to the fact that, in general, the light cone radius r increases as depth is increased, and the local dimension of the effective dynamics is $q^{O(r)}$. The latter is due to the fact that, as r increases, the number of tracked columns increases but the number of measured qudits in a single round stays constant. Hence, the fraction of measured qudits decreases, and, intuitively, we expect this to correspond to a decrease in effective measurement strength. This intuition together with the findings of prior works on unitary-and-measurement dynamics suggests that the effective dynamics experiences an entanglement phase transition from an area-law to volume-law phase as q or d is increased, and, therefore, SEBD experiences a computational phase transition, supporting Conjecture 2. While this analogy is not perfect, we provide further analytical evidence in Sec. VI that the effective 1D dynamics indeed undergoes such a phase transition.

D. Conjectured entanglement spectrum of unitary-and-measurement dynamics in an area-law phase

Numerical (Sec. V) and analytical (Sec. VI) evidence suggests that the effective 1D dynamics corresponding to the uniform 2D shallow random circuit families we

consider are in the area-law phase, making efficient simulation via SEBD very plausible. However, it is desirable to have clear predictions for the scaling of the entanglement spectra for states of the effective 1D dynamics, as this allows us to make concrete predictions for error scaling of SEBD and rule out (contrived) examples of states [93] which cannot be efficiently represented via MPS despite obeying an area law for the von Neumann entanglement entropy.

To this end, we study a simple toy model of how entanglement might scale in the area-law phase of a unitary-and-measurement circuit. Consider a chain of n qubits where we are interested in the entanglement across the cut between $1, \ldots, n/2$ and $n/2 + 1, \ldots, n$ (assume n is even). We model the dynamics as follows.

The qubits are initialized to the unentangled product state $|0\rangle^{\otimes n}$. Then, in each time step we perform the following three steps:

(1) Set the state of sites $n/2$ and $n/2 + 1$ to be an Einstein-Podolsky-Rosen (EPR) pair $|\Phi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$.

(2) Perform the cyclic permutations of qubits $(n/2, n/2 - 1, \ldots, 1)$ and $(n/2 + 1, n/2 + 2, \ldots, n)$. That is, move each qubit one step away from the central cut, except for qubits 1 and n, which are moved to $n/2$ and $n/2 + 1$, respectively.

(3) Perform a weak measurement on each qubit with Kraus elements $M_0(\theta) = \cos(\theta/2)|0\rangle\langle 0| + \sin(\theta/2)|1\rangle\langle 1|$ and $M_1(\theta) = \sin(\theta/2)|0\rangle\langle 0| + \cos(\theta/2)|1\rangle\langle 1|$. This is based on Eq. (11), but the phases do not matter here so we drop them for simplicity.

Without the measurements, this would create one EPR pair in each time step until the system has $n/2$ EPR pairs across the cut after time $n/2$. However, the measurements have the effect of reducing the entanglement. For this process, we derive the functional form of the asymptotic scaling of half-chain Schmidt coefficients $\lambda_1 \geq \lambda_2 \geq \cdots$. Moreover, bounds on the scaling of the entanglement spectrum allow us to derive a relation between the truncation error (sum of squares of discarded Schmidt values) ϵ incurred upon discarding small Schmidt values and the rank r of the post-truncation state. The bounds are given in the following lemma, which is proved in Appendix E in Supplemental Material [72].

Lemma 5.—Let $\lambda_1 \geq \lambda_2 \geq \cdots$ denote the half-chain Schmidt values after at least $n/2$ iterations of the toy model process. Then, with probability at least $1 - \delta$ the half-chain Schmidt values indexed by $i \geq i^* = \exp\{\Theta[\sqrt{\log(n/\delta)}]\}$ obey the asymptotic scaling

$$\lambda_i \propto -\exp\{\Theta[\log^2(i)]\}. \qquad (14)$$

Furthermore, upon truncating the smallest Schmidt coefficients up to a truncation error of ϵ, with probability at least

$1 - \delta$, the half-chain Schmidt rank r of the post-truncation state obeys the scaling

$$r \leq \exp\{\Theta[\sqrt{\log(n/\epsilon\delta)}]\}. \qquad (15)$$

This is the basis for our Conjecture 1′. More precisely, we take this analysis as evidence that the bond dimension D, truncation error ϵ, and system size n obey the scaling $D \leq \exp\{\Theta[\sqrt{\log(n/\epsilon\delta)}]\}$ with probability $1 - \delta$ over random circuit instance and random measurement outcomes when SEBD simulates a random constant-depth 2D circuit whose effective 1D dynamics lie in the area-law phase. Recalling that the run-time of SEBD scales like $O(nD^3)$ for a maximal bond dimension of D and using the relationship between truncation error, failure probability, variational distance error, and simulable circuit fraction given in Corollary 1, we conclude that SEBD with a maximal bond dimension cutoff scaling as $\exp\{\Theta[\sqrt{\log(n/\epsilon\delta)}]\}$ runs in time $n^{1+o(1)}\exp\{\Theta[\sqrt{\log(1/\epsilon\delta)}]\}$ and simulates $1 - \delta$ fraction of random circuit instances up to variational distance error ϵ.

It is important to note what this heuristic argument leaves out. While a 1D unitary-and-measurement circuit indeed creates $O(1)$ ebits across any given cut in each round, these do not remain in the form of distinct pairs of qubits. The unitary dynamics *within* each side of the cut have the effect of transforming the Schmidt bases into entangled ones. This makes the measurements less effective at reducing the entanglement, for reasons that can be understood in terms of quantum state merging [27,96]. Another simplification of the toy model is that the measurement angle θ is taken to be a fixed constant rather than random. Finally, in the toy model, we assume for simplicity that the EPR pairs move cyclically. We expect that, if this effect is significant, it is more likely to make the toy model overly pessimistic compared with the real situation. Despite these simplifications, we believe this model is qualitatively accurate in the area-law phase. Indeed, the scaling of Schmidt values predicted by our toy model analysis is consistent with the scaling we find numerically in Fig. 5.

IV. RIGOROUS ANALYSIS OF SEBD FOR THE EXTENDED BRICKWORK ARCHITECTURE

In this section, we show that SEBD is provably efficient for certain random circuit families that are worst-case hard. We define the circuit architecture in Fig. 3. It follows readily from prior works that exactly sampling from the output distribution of this random circuit family for arbitrary circuit instances or near-exactly computing a specific output probability with high probability is classically hard under standard complexity theoretic assumptions. We summarize these observations in the following lemma.

Lemma 6.—Let $r(L)$ and $v(L)$ be any polynomially bounded functions, with $v(L) \geq L^a$ for some $a > 0$.
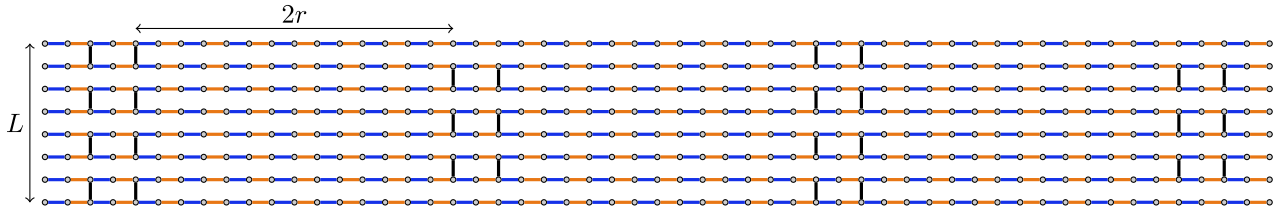
FIG. 3. Extended brickwork architecture with n qubits. Here, circles represent qubits initialized in the state $|0\rangle^{\otimes n}$, blue lines represent the first layer of gates to act, orange lines represent the second layer, and black lines represent the third and final layer. All gates are chosen Haar-randomly. We let Brickwork (L, r, v) denote the corresponding random circuit with circuit layout depicted in the figure above with vertical side length L, "extension parameter" $2r$ (which gives the distance between vertical gates acting on adjacent pairs of rows), and number of pairs of columns of vertical gates v. In the above example, $r = 7$ and $v = 4$. The standard brickwork architecture corresponds to $r = 1$. Note that $n = \Theta(Lrv)$.

Suppose that there exists a classical algorithm that runs in time poly(n) and samples from the output distribution of an arbitrary realization of Brickwork$[L, r(L), v(L)]$, as defined in Fig. 3. Then, the polynomial hierarchy collapses to the third level. Suppose there exists a classical algorithm that runs in time poly($n, 1/\delta$) and, for an arbitrary fixed output string $\mathbf{x}$, with probability at least $1 - \delta$ over choice of random instance, computes the output probability of $\mathbf{x}$ up to additive error $2^{-\tilde{\Theta}(n^2)}$. Then, there exists a probabilistic polynomial-time algorithm for computing a #P-hard function.

Proof.—We first note that Brickwork$[L, r(L), v(L)]$ supports universal MBQC, in the sense that a specific choice of gates can create a resource state that is universal for MBQC. This is an immediate consequence of the proof of universality of the "standard" brickwork architecture (corresponding to $r = 1$) proved in Ref. [90]. Indeed, when using the extended brickwork architecture for MBQC, measurements on the long 1D stretches of length $2r$ may be chosen such that the effective state is simply teleported to the end when computing from left to right (i.e., measurements may be chosen such that the long 1D segments simply amount to applications of identity gates on the effective state). The scaling $v \geq L^a$ ensures that MBQC with an extended brickwork resource state suffices to simulate any bounded-error quantum polynomial time computation with polynomial overhead. Since a specific choice of gates creates a resource state for universal MBQC, an algorithm that can simulate an arbitrary circuit realization can be used to simulate arbitrary single-qubit measurements on a resource state universal for MBQC. Under postselection, such an algorithm can, therefore, simulate post-bounded-error quantum polynomial time [85] and, hence, cannot be efficiently simulated classically unless the polynomial hierarchy collapses to the third level [8].

Similarly, for some subsets of instances, it is #P-hard to compute the output probability of an arbitrary string, since (by choosing gates to create a resource state for universal MBQC) this allows one to compute output probabilities of universal polynomial-size quantum circuit families

which are known to be #P-hard. The result of Ref. [55] is then applicable, which implies that if the gates are chosen Haar-randomly, efficiently computing the output probability of some fixed string with probability $1 - 1/\text{poly}(n)$ over the choice of instance up to additive error bounded by $2^{-\tilde{\Theta}(n^3)}$ implies the ability to efficiently compute a #P-hard function with high probability. ∎

Our goal is to prove that SEBD can efficiently approximately simulate the extended brickwork architecture in the average case for choices of extension parameters for which the above hardness results apply. To this end, we first show a technical lemma which describes how measurements destroy entanglement in 1D shallow random circuits. In particular, given a 1D state generated by a depth-2 Haar-random circuit acting on qubits, after measuring some contiguous region of spins B, the expected entanglement entropy of the resulting postmeasurement pure state across a cut going through B is exponentially small in the length of B. We defer the proof to Appendix E in Supplemental Material [72].

Lemma 7.—Suppose a 1D random circuit C is applied to qubits $\{1, \ldots, n\}$ consisting of a layer of two-qubit Haar-random gates acting on qubits $(k, k + 1)$ for odd $k \in \{1, \ldots, n - 1\}$, followed by a layer of two-qubit Haar-random gates acting on qubits $(k, k + 1)$ for even $k \in \{1, \ldots, n - 1\}$. Suppose the qubits of region $B := \{i, i + 1, \ldots, j\}$ for $j \geq i$ are measured in the computational basis, and the outcome b is obtained. Then, letting $|\psi_b\rangle$ denote the postmeasurement pure state on the unmeasured qubits, and letting $A := \{1, 2, \ldots, i - 1\}$ denote the qubits to the left of B,

$$\mathbb{E} S(A)_{\psi_b} \leq c^{|B|} \qquad (16)$$

for some universal constant $c < 1$, where the expectation is over measurement outcomes and choice of random circuit C.

We now outline the argument for why SEBD should be efficient for the extended brickwork architecture for sufficiently large extension parameters; full details may be found in Appendix E in Supplemental Material [72].

During the evolution of SEBD as it sweeps from left to right across the lattice, it periodically encounters long stretches of length $2r$ in which no vertical gates are applied. We call these "1-local regions," since the maps applied in the corresponding effective 1D dynamics are 1-local when the algorithm is in such a region. Hence, in the effective 1D dynamics, no two-qubit maps are applied, and, therefore, the bond dimension of the associated MPS cannot increase during these stretches. It turns out that in 1-local regions, not only does the bond dimension needed to represent the state not increase, but it in fact rapidly decays in expectation. If r is sufficiently large, then the effective 1D state at the end of the 1-local region is very close to a product state with high probability, regardless of how entangled the state is before the region. Hence, when SEBD compresses the MPS describing the effective state at the end of the region, it may compress the bond dimension of the MPS to some fixed constant with a very small incurred error. The two-qubit maps that are performed in between 1-local regions only increase the bond dimension by a constant factor. Hence, with high probability, SEBD can use a $O(1)$ maximal bond dimension cutoff and simulate a random circuit with extended brickwork architecture with high probability. More precisely, it turns out that the scaling $r \geq \Theta[\log(n)]$ is sufficient to guarantee efficient simulation with this argument. A more precise statement of the efficiency of SEBD for this architecture is given in the below lemma, whose proof may be found in Appendix E in Supplemental Material [72].

Lemma 8.—Let C be an instance of Brickwork (L, r, v). Then, with probability at least $1 - 2^{-\Theta(r)}$ over the circuit instance, SEBD running with maximal bond dimension cutoff $D = \Theta(1)$ and truncation error parameter $\epsilon = 2^{-\Theta(r)}$ can be used to (i) sample from the output distribution of C up to error $n2^{-\Theta(r)}$ in variational distance and (ii) compute the output probability of an arbitrary output string up to additive error $n2^{-\Theta(r)}/2^n$ in run-time $\Theta(n)$.

With an appropriate choice of $r = \Theta[\log(L)]$, the above result implies that SEBD can perform the simulation with error $1/\text{poly}(n)$ for at least $1 - 1/\text{poly}(n)$ fraction of instances. Similarly, choosing r to be a sufficiently large polynomial in L, SEBD can perform the simulation with error $2^{-n^{1-\delta}}$ for $1 - 2^{-n^{1-\delta}}$ fraction of instances, for any constant $\delta > 0$. We summarize these observations as the following corollary.

Corollary 5.—For any choice of polynomially bounded v, p_1, p_2, for any sufficiently large constant c SEBD can simulate $1 - 1/p_1(n)$ fraction of instances of Brickwork $\{L, \lceil c \log(L) \rceil, v(L)\}$ up to error $\varepsilon \leq 1/p_2(n)$ in time $O(n)$. For any choice of $\delta > 0$ and $v(L) \leq \text{poly}(L)$, for any sufficiently large constant c SEBD can simulate $1 - 2^{-n^{1-\delta}}$ fraction of instances of Brickwork $\{L, \lceil L^c \rceil, v(L)\}$ up to error $\varepsilon \leq 2^{-n^{1-\delta}}$ in time $O(n)$. Here, "simulate with error ε" implies the ability to sample with variational distance error ε and compute the output probability of some fixed string **x** with additive error $\varepsilon/2^n$.

V. NUMERICAL RESULTS

We implement SEBD on two families of random circuits: one consisting of depth-3 random circuits defined on a brickwork architecture consisting of three layers of two-qubit Haar-random gates (Fig. 3 with parameter $r = 1$) and the other being the random circuit family obtained by applying single-qubit Haar-random gates to all sites of a cluster state—we referred to this problem as CHR previously. Note that the former architecture has depth three (not including the measurement layer) and the latter has depth four, and both architectures support universal measurement-based quantum computation [90], meaning they have the worst-case-hard property relevant for Conjecture 1. We do not implement Patching, due to its larger overhead.

Implementing SEBD on a standard laptop, we could simulate typical instances of the 409×409 brickwork model with truncation error 10^{-14} per bond with a run-time on the order of one minute per sample and typical instances of the 34×34 CHR model with truncation error 10^{-10} per bond with a run-time on the order of five minutes per sample (these truncation error settings correspond to sampling errors of less than 0.01 in variational distance as derived previously in Sec. III). We, in fact, simulate instances of CHR with grid sizes as large as 50×50, although due to the significantly longer run-time for such instances we do not perform large numbers of trials for these cases. In the case of the 409×409 brickwork model, performing over 3000 trials (consisting of generating a random circuit instance and generating a sample from its output distribution using a truncation error of 10^{-14}) and finding no trials for which the bond dimension became large enough for the algorithm to fail, then with 95% confidence, we may conclude that the probability that a random trial fails, p_f, is less than 0.001. Using the bound derived in Sec. III, we can, therefore, conclude with 95% confidence that, for greater than a 0.9 fraction of 409×409 circuit instances, we can sample from that circuit instance's output distribution with variational distance error less than 0.01. Intuitively, we expect the true simulable fraction to be much larger than this statistical guarantee, as it appears that the entanglement in the effective 1D dynamics grows extensively only for highly structured instances. Note that, for both models, the run-time for a fixed truncation error is qualitatively highly concentrated around the mean. We expect that substantially larger instances of both random circuit families could be quickly simulated with more computing power, although 409×409 simulation of the brickwork architecture is already far beyond what could have been achieved by previous simulation methods that we are aware of.

To make this more precise, it is useful to compare our observed run-time with what is possible by previously known methods. The previously best-known method that we are aware of for computing output probabilities for these architectures would be to write the circuit as a tensor network and perform the contraction of the network [97]. The cost of this process scales exponentially in the tree width of a graph related to the quantum circuit [13], which, for a 2D circuit, is thought to scale roughly as the surface area of the minimal cut slicing through the circuit diagram, as in Eq. (1). Under this metric for the simulation cost, we can assert that a circuit with brickwork architecture on a 400×400 lattice using tensor network contraction would be roughly equivalent to simulating a depth-40 circuit on a 20×20 lattice with the architecture considered in Ref. [97], where the entangling gates are CZ gates. Note that general two-qubit gates can be written as a pair of tensors connected by a bond with dimension 4, but CZ gates require bond dimension only 2, meaning their contribution to the surface area is half of a random gate. With this in mind, the equivalence follows because the product of the dimensions of all the bonds crossing the minimal cut is equal to 2^{200} in both cases: For the 400×400 brickwork circuit, 100 gates cross the cut if we orient the cut horizontally through the diagram in Fig. 3 (with $r = 1$) and each gate contributes a factor of 4; meanwhile, for the depth-40 circuit on a 20×20 lattice, only one-fourth of the 40 unitary layers contain gates that cross the minimal cut, and each of these layers has 20 CZ gates that each contribute a factor of 2. The task of simulating a depth-40 circuit on a 7×7 lattice is reported to require more than two hours using tensor network contraction on the 281 petaflop supercomputer Summit [97], and the exponentiality of the run-time suggests scaling this to 20×20 would take many orders of magnitude longer, a task that is decidedly intractable.

The discrepancy between maximal lattice sizes achieved for the two architectures is a result of the fact that the two have very different effective 1D dynamics. In particular, the entanglement of the effective dynamics for the brickwork architecture saturates to a significantly smaller value than that of the cluster state architecture. And even more directly relevant for prospects of fast simulation, the typical spectrum of Schmidt values across some cut of the effective 1D dynamics for the brickwork architecture decays far more rapidly than that of the 1D dynamics for CHR. For this reason, the slower-decaying eigenvalue spectrum of CHR is significantly more costly for the run-time of the algorithm. (In fact, the eigenvalue spectrum of the brickwork model decays sufficiently quickly that we are primarily limited not by the run-time of our algorithm, but by our numerical precision, which could, in principle, be increased.) But while the slower decay of the spectrum for the CHR model necessitates a longer run-time for a given side length, it allows us to study the functional form of the spectrum and, in particular, compare against the predictions of the toy model in Sec. III D as we discuss below.

While we are computationally limited to probing low-depth and small-size models, our numerical results point toward SEBD having an asymptotic running time for both models bounded by $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$ in order to sample with variational distance ε or compute output probabilities with additive error ε/q^n with probability $1 - \delta$, suggesting that Conjecture 1 is true. Our numerical evidence for this is as follows.

(1) We find that the effective 1D dynamics associated with these random circuit families appear to be in area-law phases, as displayed in Fig. 4. That is, the entanglement does not grow extensively with the side length L but rather saturates to some constant.
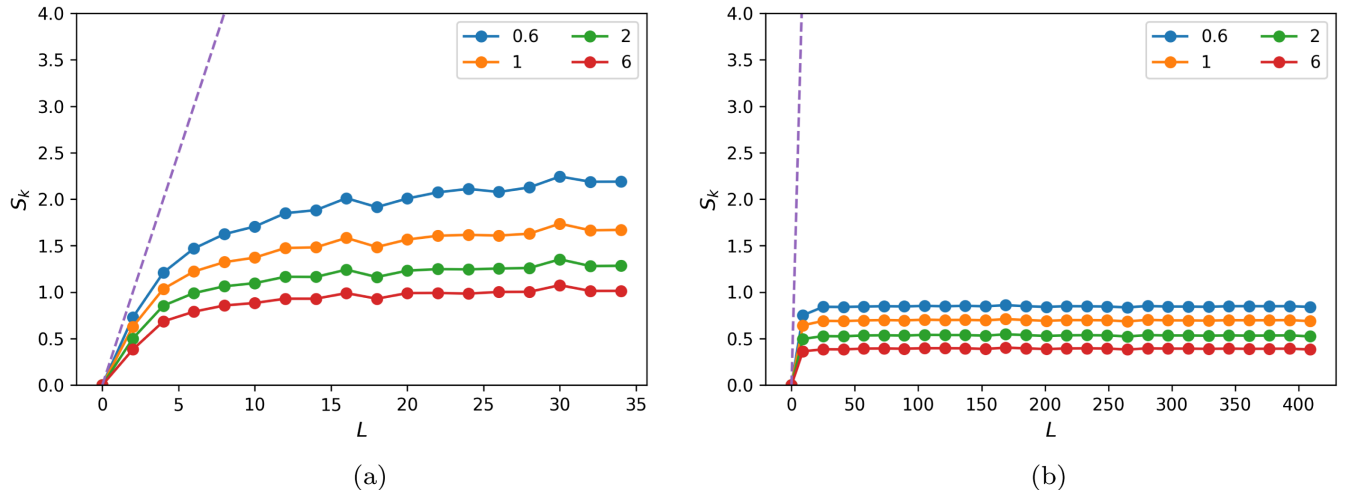

FIG. 4. Rényi half-chain entanglement entropies S_k versus side length L in the effective 1D dynamics for the CHR and brickwork models, after 80 (respectively, 550) iterations. Each point represents the entanglement entropy averaged over 50 random circuit instances and over the final ten (respectively, 50) iterations for the CHR (respectively, brickwork) model. Dashed lines depict the half-chain entanglement entropy scaling of a maximally entangled state, which can be created with a "worst-case" choice of gates for both architectures. The maximal truncation error per bond ϵ is 10^{-10} for CHR and 10^{-14} for the brickwork model. (a) CHR (b) Brickwork.

We furthermore observe qualitatively identical behavior for some Rényi entropies S_α with $\alpha < 1$. It is known [93] that this latter condition is sufficient to imply that a 1D state may be efficiently described by an MPS, indicating that SEBD is efficient for these circuit families and that Conjecture 1 is true.

(2) For further evidence of efficiency, we study the functional form of the entanglement spectra of the effective 1D dynamics. For the effective 1D dynamics corresponding to CHR, we observe superpolynomial decay of eigenvalues (i.e., squared Schmidt values) associated with some cut, displayed in Fig. 5, indicating that choosing a maximal bond dimension of $D = \mathrm{poly}(1/\epsilon)$ is more than sufficient to incur less than ϵ truncation error per bond. The observed spectrum tends toward a scaling which is qualitatively consistent with the asymptotic scaling of $\lambda_i \sim 2^{-\Theta[\log^2(i)]}$ predicted by the toy model in Sec. III D and consistent with our Conjecture 1′. Note that this actually suggests that the required bond dimension of SEBD may be even smaller than $\mathrm{poly}(1/\epsilon)$, scaling like $D = 2^{\Theta[\sqrt{\log(1/\epsilon)}]}$.

While these numerical results may be surprising given the prevalence of average-case hardness conjectures for quantum simulation, they are not surprising from the perspective of the recent works (discussed in previous sections) that find strong evidence for an entanglement phase transition from an area-law to volume-law phase for 1D unitary-and-measurement processes driven by measurement strengths. Since the effective dynamics of the 2D random shallow circuits we study are exactly such processes, our numerics simply point out that these

systems are likely on the area-law side of the transition. (However, no formal universality theorems are known, so the various models of unitary-and-measurement circuits that have been studied are generally not known to be equivalent to each other.) In the case of the brickwork architecture, we are also able to provide independent analytical evidence (Sec. VI F) that this is the case by showing the "quasientropy" $\tilde{S}_2$ for the 1D process is in the area-law phase. We leave the problem of numerically studying the precise relationship between circuit depth, qudit dimension, properties of the associated stat-mech models (including "quasientropies") as discussed in subsequent sections, and the performance of SEBD for future work. In particular, simulations of larger depth and larger qudit local dimension could be used to provide numerical support for Conjecture 2, which claims that as these parameters are increased the circuit architectures eventually transition to a regime where our algorithms are no longer efficient.

VI. ANALYTICAL EVIDENCE FOR CONJECTURES FROM STATISTICAL MECHANICS

A. Overview

In the previous section, we provide strong numerical evidence that SEBD is efficient when acting on certain sufficiently shallow architectures. Here, we provide complementary, analytical evidence that bolsters the case for SEBD's (and, in Appendix C in Supplemental Material [72], Patching's) efficiency. The method is based on a technique developed in Refs. [29,30,98–101] that maps random quantum circuits to classical statistical mechanical
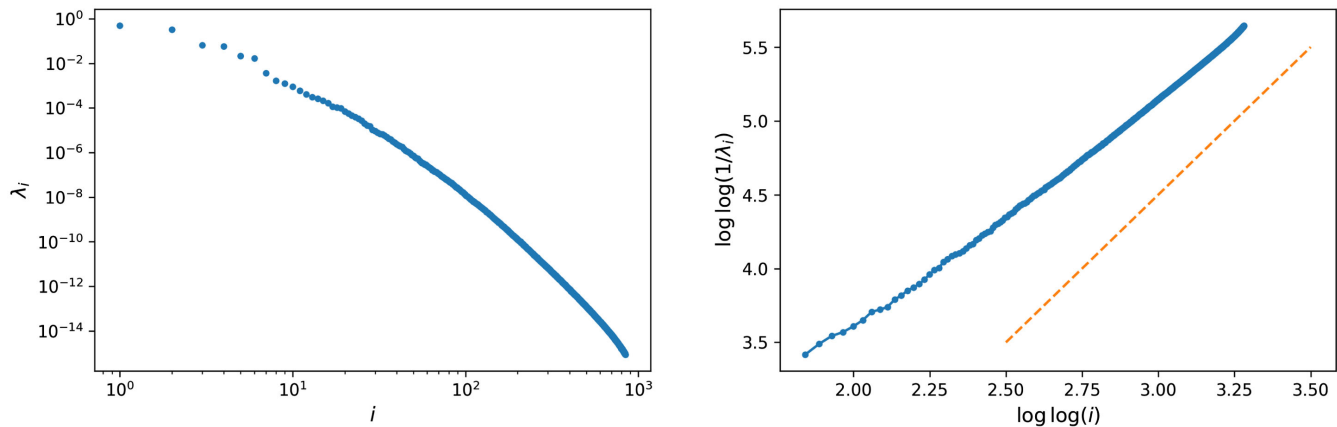

FIG. 5.   Typical half-chain entanglement spectrum $\lambda_1 \geq \lambda_2 \geq \cdots$ observed during the effective 1D dynamics of CHR. These plots are generated from an instance with side length $L = 44$ after running for 44 iterations, with squared Schmidt values smaller than approximately 10^{-15} truncated. The left figure shows a spectrum of half-chain eigenvalues. The downward curvature in the log-log scale indicates superpolynomial decay. The right figure displays the same data (minus the few largest values) on a loglog-loglog scale. The toy model predicts that the blue curve asymptotes to a straight line with slope two in the right figure, illustrated by the dashed orange line, corresponding to scaling like $\lambda_i \sim 2^{-\Theta[\log^2(i)]}$. The plot is qualitatively consistent with this prediction. The spectrum for the brickwork model decays too quickly to obtain as useful statistics without going to much higher numerical precision.

models. We describe how the method can be applied generally to different 2D architectures, but we give special attention to the depth-3 brickwork architecture, because it is a worst-case-hard uniform architecture which is simple enough for concrete conclusions to be drawn that act as evidence that the algorithms are efficient. The stat-mech mapping also provides evidence of computational phase transitions as qudit dimension and circuit depth are increased.

The mapping produces a classical stat-mech model for which the entanglement properties of the underlying random circuit are related to thermodynamic properties of the model. In particular, we examine a quantity we call the "quasi-k entanglement entropy" $\tilde{S}_k$ to quantify the entanglement of the 1D state "tracked" by SEBD at any given point in time throughout the effective 1D dynamics; the mapping relates $\tilde{S}_k$ to the free energy cost incurred by twisting boundary conditions of the stat-mech system. The expression for quasi-k entropy resembles that of the Rényi-k entanglement entropy averaged over random circuit instances and measurement outcomes, denoted by $\langle S_k \rangle$, and exactly yields the expected von Neumann entropy in the limit of $k \to 1$. To rigorously prove that SEBD is efficient, it would suffice to upper bound $S_k \leq O(\log n)$ for any $0 < k < 1$. [102] To show that the algorithm is *inefficient*, it would suffice to *lower bound* $S_k \geq \Omega(n^{\Theta(1)})$ for any $k \geq 1$. These entropy bounds need to hold for the majority of randomly chosen circuit instances, and, in case the entropy varies throughout the algorithm, a proof of efficiency would need low entropy for all states across all cuts, and a proof of inefficiency would need high entropy at only one point.

The stat-mech mapping we use gives us access to the quasientropies $\tilde{S}_k$ for integer $k \geq 2$. The calculations are especially tractable for $k = 2$, and this mostly suffices for higher k, since $S_2 \geq S_k \geq [S_2/(k-1)]$ for $k > 1$. While examples exist where S_k for $k < 1$ can differ dramatically from S_2, our numerical studies described in the previous section do not find large differences between these quantities.

Changing the qudit dimension q of the random circuit model corresponds to changing the interaction strengths in the associated stat-mech model, which drives a phase transition. We see that the ordered (disordered) phases of the stat-mech model correspond to volume-law (area-law) entanglement for $\tilde{S}_2$ in the effective 1D dynamics. According to the above discussion, this suggests but does not prove that SEBD is inefficient (efficient) when the stat-mech model is ordered (disordered).

In the remaining subsections, we define the quasientropy, explain the stat-mech map (with special attention for the case of $k = 2$), apply it generally to 2D circuits to reason heuristically about order-disorder behavior, and finally conclude by applying it more rigorously to the depth-3 brickwork architecture, where we observe

a q-driven order-disorder phase transition in the corresponding stat-mech model. A more general and more detailed formulation of the stat-mech mapping, including its mathematical justification, is given in Appendix A in Supplemental Material [72].

B. Quasientropy

Given an ensemble of pure quantum states, the quasientropy is a quantity that is related to the expected amount of entanglement in the state. In our case, the ensemble is generated by a random quantum circuit followed by a projective measurement on some subset of the qudits, and the quasientropy is computed as follows.

Suppose we fix a random quantum circuit instance drawn according to some specified architecture, as well as a known outcome for a projective measurement performed on some subset of the output qudits. Let ρ be the pure output state on the unmeasured qudits associated with the instance and measurement outcome, and fix the normalization $\mathrm{Tr}(\rho)$ to be equal to the probability of obtaining the specified measurement outcome. Then, for any $k \geq 0$ and for some subregion A of the unmeasured qudits, we define

$$Z_{k,\varnothing} = \mathrm{tr}(\rho)^k, \tag{17}$$

$$Z_{k,A} = \mathrm{tr}(\rho_A^k), \tag{18}$$

where ρ_A is the reduced density matrix of ρ on region A. Letting $\mathbb{E}_U$ denote expectation over choice of instance and uniformly random measurement outcome, the quasi-k entropy $\tilde{S}_k(A)$ for the random circuit ensemble is defined as

$$\tilde{S}_k(A) := \frac{1}{1-k} \log\left(\frac{\mathbb{E}_U\left[\mathrm{tr}(\rho)^k \frac{Z_{k,A}}{Z_{k,\varnothing}}\right]}{\mathbb{E}_U[\mathrm{tr}(\rho)^k]} \right) \tag{19}$$

$$= \frac{1}{1-k} \log\left(\frac{\mathbb{E}_U(Z_{k,A})}{\mathbb{E}_U(Z_{k,\varnothing})} \right) \tag{20}$$

$$= \frac{F_{k,\varnothing} - F_{k,A}}{1-k}, \tag{21}$$

where $F_{k,X} := -\log[\mathbb{E}_U(Z_{k,X})]$ for $X \in \{\varnothing, A\}$ is associated with the "free energy" of the classical stat-mech model that the circuit maps to. Virtually identical quantities are also considered in two other recent works [29,30].

Note the similarity of the above expression to the average Rényi-k entanglement entropy:

$$\langle S_k(A)_\rho \rangle := \frac{\mathbb{E}_U[\mathrm{tr}(\rho)S_k(A)_\rho]}{\mathbb{E}_U[\mathrm{tr}(\rho)]} \tag{22}$$

$$= \frac{1}{1-k} \frac{\mathbb{E}_U[\mathrm{tr}(\rho)\log\frac{Z_{k,A}}{Z_{k,\varnothing}}]}{\mathbb{E}_U[\mathrm{tr}(\rho)]}. \tag{23}$$

Indeed, the two formulas are the same, except that the quasientropy weights instances by $\text{Tr}(\rho)^k$ instead of $\text{Tr}(\rho)$ and takes the logarithm after taking the expectation.

Also note that, in the limit $k \to 1$, both $\tilde{S}_k$ and $\langle S_k \rangle$ approach the expected von Neumann entropy

$$\langle S(A) \rangle = -\mathbb{E}_U \left\{ \text{tr} \left[\frac{\rho_A}{\text{tr}(\rho)} \log \left(\frac{\rho_A}{\text{tr}(\rho)} \right) \right] \right\}. \quad (24)$$

This limit is a manifestation of the *replica trick* [29,30,100], where $\tilde{S}_k$ is computed for integer values of $k \geq 2$ and $\langle S_1 \rangle$ is subsequently inferred by analytic continuation to $k = 1$. By formally connecting $\tilde{S}_k$ to more meaningful entropic quantities, this observation justifies our study of $\tilde{S}_k$ as an indicator of the efficiency of our algorithms.

C. Mapping

We now describe the procedure for mapping a random quantum circuit family to a classical statistical mechanical model, such that quantities $\mathbb{E}_U(Z_{k,\varnothing})$ and $\mathbb{E}_U(Z_{k,A})$ for integers $k \geq 2$ are given by partition functions of the stat-mech model. This follows work in Refs. [29,30,98–101], although our presentation is for the most part self-contained. Here, we present merely how to perform the mapping, leaving the details of its justification to Appendix A in Supplemental Material [72]. In that Appendix, we also present a more generalized version of the mapping that accounts for the possibility of weak measurements acting in between Haar-random gates.

To define the stat-mech model, we must specify two ingredients: first, the nodes and edges that form the interaction graph on which the model lives and, second, the details of the interactions between nodes that share an edge. The graph, which is the same for all k, is formed from the circuit diagram as follows. First, we replace each Haar-random unitary (labeled by integer u) in the circuit diagram
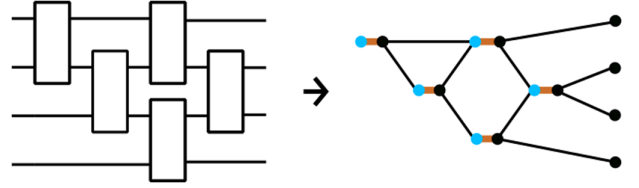
with a pair of nodes, which we refer to as the *incoming* node t_u and *outgoing* node s_u for that unitary, and we connect nodes t_u and s_u by an edge. Then, we add edges between the outgoing node s_{u_1} of unitary u_1 and the incoming node t_{u_2} of another unitary u_2 when u_2 acts immediately after u_1 on the same qudit. Finally, we introduce a single auxiliary node x_a for each qudit $a \in \{1, 2, \ldots, n\}$ that is not measured (recall ρ is the output only on the unmeasured qubits), and we add a single edge connecting x_a to the outgoing node s_u for unitary u if u is the final unitary of the circuit to act on qudit a. Thus, all of the incoming and outgoing nodes have degree equal to three, unless they are associated with the first unitary to act on a certain qubit or the last unitary to act on a measured qubit. We provide a simple example of this mapping in Fig. 6.

Each node in the graph may now be viewed as a spin that takes on one of $k!$ values, corresponding to an element of the symmetric group S_k. A spin configuration is given by an assignment $(\sigma_u, \tau_u) \in S_k \times S_k$ for each pair of nodes (s_u, t_u), as well as an assignment $\chi_a \in S_k$ to each auxiliary node x_a. The main utility of the stat-mech mapping is then given by the following equation, expressing the quantity $\mathbb{E}_U(Z_{k,X})$ for $X = \varnothing$ or $X = A$ as a sum over spin configurations on this graph:

$$\mathbb{E}_U(Z_{k,X}) = \sum_{\{\sigma_u\}_u, \{\tau_u\}_u} \prod_u \text{weight}(\langle s_u t_u \rangle) \prod_{\langle s_{u_1} t_{u_2} \rangle} \text{weight}(\langle s_{u_1} t_{u_2} \rangle) \prod_{\langle s_u x_a \rangle} \text{weight}(\langle s_u x_a \rangle). \quad (25)$$

This is a partition function—a weighted sum over spin configurations where the weight of each term is given by a product of factors that depend only on the spin value of a pair of nodes (s, t) connected by an edge, denoted $\langle st \rangle$. In this case, the sum runs only over the values σ_u and τ_u, of the incoming and outgoing nodes; the values χ_a of the auxiliary nodes are fixed across all the terms and encode the boundary conditions that differ between $\mathbb{E}_U(Z_{k,\varnothing})$ and $\mathbb{E}_U(Z_{k,A})$. We define the free energy to be the negative logarithm of this partition function [see Eq. (21)], mirroring the standard relationship $F = -k_B T \log(\tilde{Z})$ between the

free energy and the partition function from statistical mechanics, with $k_B T$ set to 1.

We now specify the details of the interaction by defining the weight function for different edges. There are only two different kinds of interactions. Edges $\langle s_u t_u \rangle$ between incoming and outgoing nodes of the same unitary have

$$\text{weight}(\langle s_u t_u \rangle) = wg(\tau_u \sigma_u^{-1}, q^2), \quad (26)$$

where $wg(\pi, q^2)$ is the Weingarten function. The Weingarten function arises from performing the

expectations over the Haar measure in evaluation of the expressions for $Z_{k,\varnothing}$ and $Z_{k,A}$, and one formula for it is given in Supplemental Material [72] in Eq. (A19). Note that there exist permutations π for which $wg(\pi, q^2) < 0$, so the overall weight of a configuration can be negative and our stat-mech model corresponds only to a physical model with complex-valued energy.

Meanwhile, edges $\langle s_{u_1} t_{u_2} \rangle$ connecting nodes of successive unitaries u_1 and u_2 (respectively, edges $\langle s_u x_a \rangle$ connecting outgoing nodes to auxiliary nodes) have weight that depends only on the number of cycles in the product of the permutations assigned to each of the nodes:

$$\text{weight}(\langle s_{u_1} t_{u_2} \rangle) = q^{C(\sigma_{u_1} \tau_{u_2}^{-1})}, \tag{27}$$

$$\text{weight}(\langle s_u x_a \rangle) = q^{C(\sigma_u \chi_a^{-1})}, \tag{28}$$

where $C(\pi)$ returns the number of cycles that make up the permutation π. This weight function becomes more complicated when weak measurements are applied in between gates u_1 and u_2, a generalization we discuss further in Appendix A in Supplemental Material [72].

The final piece of this prescription is setting the value χ_a for each of the auxiliary nodes x_a at the end of the circuit, which can be seen as fixing the boundary conditions for the stat-mech model. These nodes are fixed to the same value for each term in the sum and depend on whether we are calculating $\mathbb{E}_U(Z_{k,\varnothing})$ or $\mathbb{E}_U(Z_{k,A})$ and whether the qudit a is in the region A. For $\mathbb{E}_U(Z_{k,\varnothing})$, the value χ_a is fixed to the identity permutation e for every a. Meanwhile, for $\mathbb{E}_U(Z_{k,A})$, we "twist" the boundary conditions and change χ_a to be the k-cycle $(1\ldots k)$ if a is in A, leaving $\chi_a = e$ if a is in the complement of A.

D. Special case of $k = 2$

When $k = 2$, the symmetric group S_k has only two elements, identity (denoted by e) and swap [denoted by (12) in cycle notation], so the quantities $\mathbb{E}_U(Z_{2,\varnothing})$ and $\mathbb{E}_U(Z_{2,A})$ map to partition functions of Ising-like classical stat-mech models where each node takes on one of two values. Furthermore, in the $k = 2$ case with no measurements, it is shown in Refs. [98,99] (see also Refs. [100,101]) that one can get rid of all negative terms in the partition function by decimating half of the nodes, i.e., explicitly performing the sum over the values of the incoming nodes $\{\tau_u\}_u$ in Eq. (25). This continues to be true even when there are measurements in between unitaries in the circuit, as discussed in Supplemental Material [72]. However, the decimation causes the two-body interactions to become three-body interactions between any three nodes s_{u_1}, s_{u_2}, and s_{u_3} when unitary u_3 succeeds unitaries u_1 and u_2 and shares a qudit with each. This interaction is intrinsically three-body; if one attempts to decompose it into three two-body interactions, they find these interactions

need to have infinite strength [see Eqs. (A37) and (A38) in Supplemental Material [72] with the substitution $w = 1/q$]. The lack of negative weights for $k = 2$ is convenient, because it allows one to view the system as a classical spin model at a real temperature and can, therefore, be analyzed with well-studied numerical techniques like Monte Carlo sampling.

E. Mapping applied to general 2D circuits

We now apply the mapping directly to a depth-d circuit acting on a $\sqrt{n} \times \sqrt{n}$ lattice of qudits consisting of nearest-neighbor two-qudit Haar-random gates. This is the relevant case for the algorithms presented in this paper. In this section, we assume for concreteness that the first unitary layer includes gates that act on qudits at grid points (i, j) and $(i, j + 1)$ for all odd i and all j, the second layer on (i, j) and $(i, j + 1)$ for all even i and all j, the third layer on (i, j) and $(i + 1, j)$ for all i and all odd j, and the fourth layer on (i, j) and $(i + 1, j)$ for all i and all even j. Subsequent layers then cycle through these four orientations.

1. The classical stat-mech model

Replacing the unitaries in the circuit diagram with pairs of nodes and connecting them as described previously yields a graph embedded in three dimensions. The nodes in this graph still have degree three, so locally the graph looks similar to the honeycomb lattice (the lattice that arises from a $1 + 1$D circuit as discussed in Refs. [29,30,98,101] and in Supplemental Material [72]), but globally the nodes form a 3D lattice that can be viewed roughly as a $\sqrt{n} \times \sqrt{n} \times d$ slab, although the details of how these nodes connect is not straightforward to visualize. We include pictures of the graph in Fig. 7.

Recall that edges between nodes originating from the same unitary are assigned a weight equal to the Weingarten function and edges between successive unitaries follow the

(a) Depth-4 circuit on 4×4 lattice
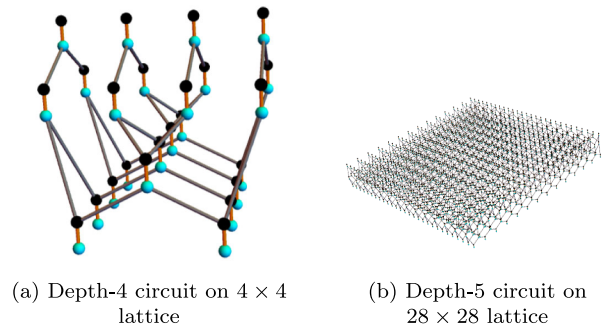
(b) Depth-5 circuit on 28×28 lattice

FIG. 7. The graph produced by the stat-mech mapping on shallow 2D circuits. (a) A close-up view of the graph reveals that the degree of most nodes is three, similar to the honeycomb lattice. (b) A far-away view reveals that globally the graph looks like a two-dimensional slab of thickness roughly d.

interaction weight($\langle s_{u_1} t_{u_2} \rangle) = q^{C(\sigma_{u_1} \tau_{u_2}^{-1})}$. For $k = 2$ this amounts to a ferromagnetic Ising interaction where

$$\text{weight}(\langle s_{u_1} t_{u_2} \rangle) = \begin{cases} q^2 & \text{if } \sigma_{u_1} \tau_{u_2} = e, \\ q & \text{if } \sigma_{u_1} \tau_{u_2} = (12). \end{cases} \quad (29)$$

To analyze the output state, we divide the n qudits into three groups A, B, and C. We suppose that, after the d unitary layers are performed, a projective measurement is performed on the qudits in region B. Qudits in regions A and C are left unmeasured, and we wish to calculate quantities like $\mathbb{E}_U(Z_{k,\varnothing})$ and $\mathbb{E}_U(Z_{k,A})$. The mapping calls for us to introduce an auxiliary node for each unmeasured qudit in the circuit, i.e., an auxiliary node for qudits in regions A and C. For $\mathbb{E}_U(Z_{k,\varnothing})$, all of the auxiliary nodes are set to identity e, while for $\mathbb{E}_U(Z_{k,A})$, the auxiliary nodes for region A are set to the k-cycle $(1\ldots k)$.

2. Eliminating negative weights via decimation when k = 2

The quantities $\mathbb{E}_U(Z_{k,\varnothing/A})$ are now given by classical partition functions on this graph with appropriate boundary conditions for the auxiliary nodes in regions A and C. We wish to understand whether this stat-mech model is ordered or disordered. We are faced with the issue that the Weingarten function can take negative values and, thus, some configurations over this graph could have negative weight. For $k = 2$, as previously discussed, we can rectify this by decimating all the incoming nodes. The resulting graph has half as many nodes and interactions between groups of three adjacent nodes s_{u_1}, s_{u_2}, and s_{u_3}, whenever unitary u_3 acts after u_1 and u_2. There is a simple formula for the weights:

$$\text{weight}(\langle s_{u_1} s_{u_2} s_{u_3} \rangle) = \begin{cases} 1 & \text{if } \sigma_{u_1} = \sigma_{u_2} = \sigma_{u_3}, \\ \frac{1}{q + q^{-1}} & \text{if } \sigma_{u_2} \neq \sigma_{u_3}, \\ 0 & \text{if } \sigma_{u_1} \neq \sigma_{u_2} = \sigma_{u_3}. \end{cases}$$
$$(30)$$

Now, all the weights are non-negative. Moreover, the largest weight occurs when all the nodes agree, indicating a generally ferromagnetic interaction between the trio of nodes. If either σ_{u_1} or σ_{u_2} disagrees with the other two values, the weight is reduced by a factor of $q + q^{-1}$. When σ_{u_3} disagrees, the weight is 0; these configurations are forbidden and contribute nothing to the partition function.

Given an assignment of e or (12) to each node σ_u, we can associate a pattern of domain walls, that is, a set of edges connecting nodes with disagreeing values. These domain walls partition the 2D slab into contiguous domains of adjacent nodes all given the same value.

3. Allowed domain wall configurations and disorder-order phase transitions

Using this observation, we can understand the kinds of domain wall structures that appear in configurations that contribute nonzero weight. Recall that the stat-mech model occupies a 2D slab of constant thickness in the direction of time, which we orient vertically. In this setting, domain wall structures are membranelike, since the graph is embedded in 3D. Membranes that have upward curvature, shaped like a bowl, are not allowed, because somewhere there would need to be an interaction where the upper node disagrees with the two below it, a situation that leads to 0 weight as in Eq. (30). On the other hand, cylindrically shaped domain wall membranes do not have this issue, nor do dome-shaped membranes with downward curvature. These three cases are illustrated in Fig. 8. The weight of a configuration is reduced by a factor of $q + q^{-1}$ for each unit of domain wall, an effect that acts to minimize the domain wall size when drawing samples from the thermal distribution (energy minimization). On the other hand, larger domain walls have more configurational entropy—there are many ways to cut through the graph with a cylindrically shaped membrane—an effect that acts to bring out more domain walls in samples from the thermal distribution (entropy maximization). The question is, which of these effects dominates? For a certain setting of the depth d (slab thickness) and local dimension q, is there long-range order, or is there exponential decay of correlations indicating disorder? Generally speaking, increasing depth magnifies both effects: Cylindrical domain wall membranes must be
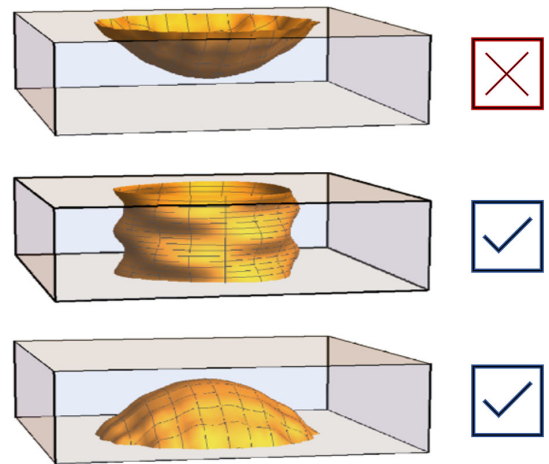

FIG. 8. Cartoon depiction of forbidden and allowed domain wall structures in the stat-mech model for a shallow 2D circuit. Time is oriented vertically. For a particular spin configuration, domain walls mark the boundary between regions assigned e and regions assigned (12), forming a membrane. If this membrane has upward curvature, it is forbidden (contributes 0 weight to the partition function), whereas if it does not have upward curvature, it is allowed. For allowed configurations, the contribution decreases exponentially in the total domain wall area.

longer—meaning larger energy—when the depth is larger; however, longer cylinders also have more ways of propagating through the graph. Meanwhile, increasing q only magnifies the energetic effect, since it increases the interaction strength and, thus, the energy cost of a domain wall unit but leaves the configurational entropy unchanged.

Thus, in the limit of large q we expect the energetic effect to win out and the system to be ordered for any fixed circuit depth d and any circuit architecture. What about small q? Physically speaking, q must be an integer at least 2, since it represents the local Hilbert space dimension of the qudit. However, the statistical mechanical model itself requires no such restriction, and we can allow q to vary continuously in the region $[1, \infty)$. Then, for $q \to 1$, the energy cost of one unit of domain wall becomes minimal (but it does not vanish). Depending on the exact circuit architecture and the depth of the circuit, the system may experience a phase transition into the disordered phase once q falls below some critical threshold q_c. The depth-3 circuit with brickwork architecture that we present later in Sec. VI F provides an example of such a transition that can be more explicitly analyzed, because it can be directly related to the well-studied 2D Ising model. It is disordered when $q = 2$ and experiences a phase transition as q increases to the ordered phase at a transition point we estimate to be roughly $q_c \approx 6$.

When q is fixed and d is varied, it is less clear what to expect. Suppose for small d the system is disordered. Then, increasing d amplifies both the energetic and entropic effects but likely not in equal proportions. If the amplification of the energetic effect is stronger with increasing depth, then we expect to transition from the disordered phase to the ordered phase at some critical value of the depth d_c. Without a better handle on the behavior of the stat-mech model, we cannot definitively determine if and when this depth-driven phase transition happens.

However, we have other reasons to believe that there should be a depth-driven phase transition. In particular, we now provide an intuitive argument for why a disorder-order transition in the parameter q should imply a disorder-order transition in the parameter d. Consider fixed d and another fixed integer $r \geq 1$ such that $d/r \gg 1$. We may group together $r \times r$ patches of qudits to form a "supersite" with local dimension q^{r^2}. Similarly, we may consider a "superlayer" of $O(r)$ consecutive unitary layers. Since $O(r)$ layers are sufficient to implement an approximate unitary k design on a $r \times r$ patch of qudits [taking $k = O(1)$] [52], we intuitively take each superlayer to implement a Haar-random unitary between pairs of neighboring supersites. Thus, a depth-d circuit acting on qudits of local dimension q is roughly equivalent to a depth-$O(d/r)$ circuit acting on qudits of local dimension q^{r^2} in the supersite picture. If, for a fixed d, we observe a disorder-order phase transition for increasing q, then, for fixed q and fixed d/r, we should also observe a disorder-order phase transition with increasing r. Equivalently, we should see a transition for fixed q and

increasing d. This logic is not perfect, because superlayers do not exactly map to layers of Haar-random two-qudit gates between neighboring supersites, but nonetheless we take it as reason to expect a depth-driven phase transition.

4. Efficiency of SEBD algorithm from stat mech

The efficiency of the SEBD algorithm relies on the error incurred during the MPS compression being small. If the inverse error has a polynomial relationship (or better) with the bond dimension of truncation, then the algorithm's time complexity is polynomial (or better) in the inverse error and the number of qudits. This is the case if the MPS prior to truncation satisfies an area law for the k-Rényi entropy for some $0 < k < 1$. The stat-mech mapping is unable to probe these values of k. However, we hypothesize that the behavior of larger values of k is indicative of the behavior for $k < 1$, since the examples where the k-Rényi entropy with $k \geq 1$ satisfies an area law but efficient MPS truncation is not possible require contrived spectrums of Schmidt coefficients. Although some physical processes give rise to situations where the von Neumann and k-Rényi entropies with $k > 1$ exhibit different behavior [see, e.g., Ref. [103], which shows that, for random 1D circuits without measurements but with the unitaries chosen to commute with some conserved quantity, after time t the entropy is $O(t)$ for $k = 1$ but $O(\sqrt{t \log t})$ for $k > 1$], the numerical evidence we give in Sec. V, where the scaling of all the k-Rényi entropies appears to be the same, suggests our case is not one of these situations.

Previously, we discussed how for 1D circuits with alternating unitary and weak measurement dynamics there has been substantial numerical evidence in prior literature for a phase transition from an area-law phase to a volume-law phase as the parameters of the circuit are changed. There has also been analytic work [29,30] on this model using the stat-mech mapping (and, in Appendix A in Supplemental Material [72], we use a similar approach to analyze 1D circuits with a different form of weak measurement, inspired by the CHR problem discussed earlier, and show there is a q-driven phase transition from a disordered phase to an ordered phase).

The SEBD algorithm simulating a 2D circuit of constant depth made from Haar-random gates may be viewed as a system with very similar dynamics—an alternation between entanglement-creating unitary gates and entanglement-destroying weak measurements. However, none of the unitary-and-measurement models that have been previously studied capture the exact dynamics of SEBD, one reason being that SEBD tracks the evolution of several columns of qudits at once (recall it must include all qudits within the light cone of the first column). The Haar-random unitaries create entanglement within these columns of qudits but not in the exact way that entanglement is created by Haar-random nearest-neighbor gates acting on a single column. Nonetheless, we expect the story to be the same for
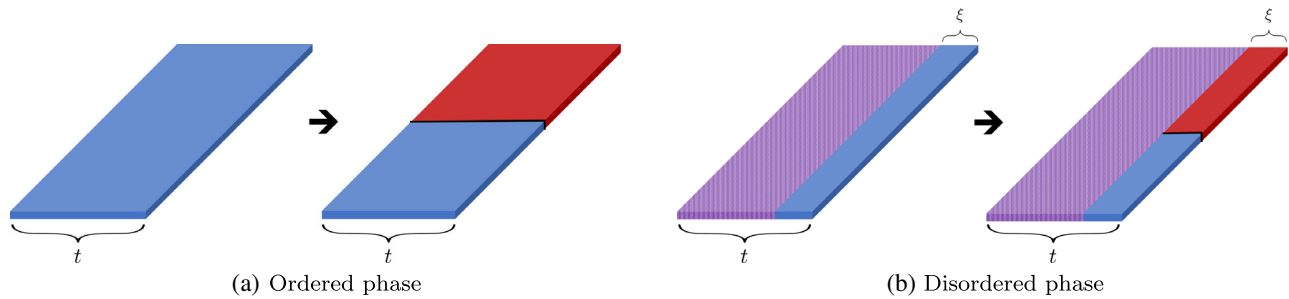
(a) Ordered phase

(b) Disordered phase

FIG. 9. The stat-mech mapping yields nodes arranged within a roughly $\sqrt{n} \times t \times d$ prism. (a) In the ordered phase, twisting the boundary conditions at the right boundary introduces a domain wall between the two phases (indicated by red and blue) that propagates through the bulk for a total area of $O(td)$. (b) In the disordered phase, boundary conditions introduce bias that is noticeable only within a constant $O(\xi)$ distance of the boundary, and the domain wall membrane introduced by twisting the boundary conditions is quickly washed out by the bulk disorder (dotted purple). The total area is $O(\xi d)$.

the dynamics of SEBD, since the main findings of studies of these unitary-and-measurement models are quite robust to variations in which unitary ensembles and which measurements are being implemented; we expect that varying parameters of the circuit architecture like q and d can lead to entanglement phase transitions and, thus, transitions in computational complexity.

Indeed, the discussion from the previous section suggests precisely this fact. When we apply the stat-mech mapping directly to 2D circuits instead of to 1D unitary-and-measurement models, we expect disorder-order phase transitions as both q and d are varied. To make the connection to entanglement entropy explicit here, we note that, after t steps of the SEBD algorithm, all $\sqrt{n}$ qudits in the first t columns of the $\sqrt{n} \times \sqrt{n}$ lattice are measured, and we have an MPS representation of the state on columns $t+1$ through $t+r$, where $r = O(d)$ is the radius of the light cone (which depends on circuit architecture but cannot be larger than d). To calculate the entropy of the MPS, we take the region A to be the top half of these r columns and region C to be the bottom half. Region B consists of the first t columns, which experience projective measurements. The prescription for computing $\tilde{S}_2(A)$ calls for determining the free energy cost of twisting the boundary conditions in region A, which creates a domain wall along the $A:C$ border. If the bulk is in the ordered phase, then this domain wall membrane originating at the $A:C$ boundary penetrates through the graph a distance of t, leading to a domain wall area of $O(td)$. If the bulk is in the disordered phase, it penetrates only a constant distance, on the order of the correlation length ξ of the disordered stat-mech model, before being washed out by the disorder, leading to a domain wall area of only $O(\xi d)$. This is the key observation that connects order-disorder to the quasientropy; the observation is inspired by a similar transition for random tensor networks (as opposed to random quantum circuits), studied in Ref. [104]. The typical domain wall configurations before and after twisting boundary conditions in the ordered and disordered phases are reflected in the cartoon in Fig. 9. As elaborated upon in Appendix A in

Supplemental Material [72], we expect there to be a correspondence between the scaling of the domain wall size and the free energy cost after twisting the boundary conditions of the stat-mech model.

This implies that the quasientropy $\tilde{S}_2$ is in the area-(respectively, volume-) law phase when the classical stat-mech model is in the disordered (respectively, ordered) phase. Heuristically, we might expect the run-time of the SEBD algorithm to scale like $\text{poly}(n)\exp[O(\tilde{S}_2)]$, suggesting that the disorder-to-order transition is accompanied by an efficient-to-inefficient transition in the complexity of the SEBD algorithm. Furthermore, near the transition point within the volume-law phase, the quasientropy scales linearly with system size but with a small constant prefactor, suggesting that the SEBD run-time, though exponential, could be considerably better than previously known exponential-time techniques.

F. Depth-3 2D circuits with brickwork architecture

Now, we turn our attention specifically to the depth-3 brickwork architecture that we also numerically simulate. In this architecture, three layers of two-qudit gates are performed on a 2D lattice of qudits as shown in Fig. 10(a). Note that this architecture is also introduced in Sec. IV; the architecture we consider here is exactly the extended brickwork architecture of that section with the extension parameter r fixed to be one.

As previously discussed in Sec. IV, this structure is known to be universal in the sense that one may simulate any quantum circuit using a brickwork circuit (with polynomial overhead in the number of qudits) by judiciously choosing which two-qudit gates to perform and performing adaptive measurements [90]. Thus, it is hard to exactly sample or compute the output probabilities of brickwork circuits in the worst case assuming the polynomial hierarchy does not collapse, and we expect neither the SEBD algorithm nor the Patching algorithm to be efficient. However, we now give evidence that these algorithms are efficient in the "average case," where each two-qudit gate is Haar random, by considering the order
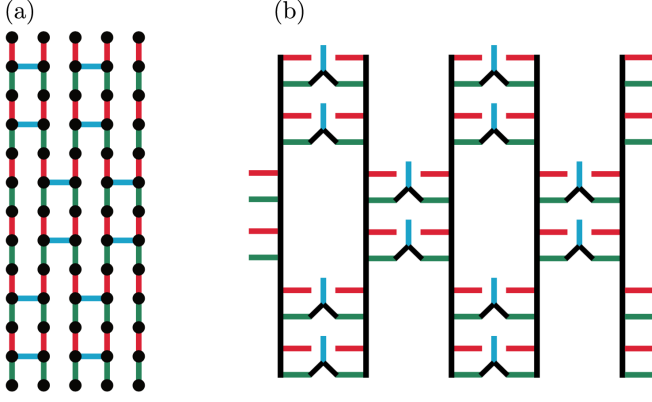
FIG. 10. (a) Brickwork architecture. Qudits lie at the location of black dots. Three layers of two-qudit gates act between nearest-neighbor qudits—first qudits linked by a vertical red edge, then vertical green, and then horizontal blue. In our SEBD simulation of this circuit architecture, we sweep from left to right. (b) Result of stat-mech mapping applied to brickwork architecture depicted in (a). Nodes are implied to lie at the end points of each edge. Red, green, and blue edges carry Weingarten weight. Black edges carry weight given by $\text{weight}(\langle s_{u_1} t_{u_2} \rangle) = q^{C(\sigma_{u_1} \tau_{u_2}^{-1})}$.

and disorder properties of the stat-mech model that the brickwork architecture maps to.

1. Stat-mech mapping for general k

The stat-mech mapping proceeds as previously discussed for 2D circuits, but we see that the brickwork architecture allows us to make some important simplifications. Each gate in the circuit is replaced by a pair of nodes, which are connected with an edge. Then, the outgoing nodes of the first (red) layer are connected to the incoming nodes of the second (green) layer, and the outgoing nodes of the second (green) layer are connected to the incoming nodes of the third (blue) layer. The resulting graph is shown in Fig. 10(b). Edges connecting incoming and outgoing nodes of the same layer are shown in color (red, green, and blue) and carry weight equal to the Weingarten function. Edges connecting subsequent layers are black. These edges carry weight given by $\text{weight}(\langle s_{u_1} t_{u_2} \rangle) = q^{C(\sigma_u \tau_{u_2}^{-1})}$.

To perform the full mapping, we also add a layer of auxiliary nodes for any unmeasured qudits and connect them to the third layer. However, we are interested primarily in the bulk order-disorder properties of the system and suppose that all the qudits, except perhaps those at the boundary of the 2D system, are measured after the third layer, so we need not consider auxiliary nodes.

Looking at Fig. 10(b), we see that some of the nodes have degree 1 and connect to the rest of the graph via a (red or blue) Weingarten link. We can immediately decimate these nodes from the graph. For any τ, we have [105]

$$\sum_{\sigma \in S_k} wg(\tau\sigma^{-1}, q^2) = \sum_{\sigma \in S_k} wg(\sigma, q^2) = \frac{(q^2-1)!}{(k+q^2-1)!}, \quad (31)$$

which is independent of τ, so decimating these spins merely contributes the above constant to the total weight. This constant appears in both the numerator and denominator of quantities like $\mathbb{E}_U(Z_{k,A})/\mathbb{E}_U(Z_{k,\varnothing})$, and we ignore them henceforth. The remaining graph can be straightened out, yielding Fig. 11(a). The fact that Fig. 11(a) is a graph embedded in a plane that includes only two-body interactions is one upshot of studying the brickwork architecture, as it makes the analysis more straightforward and the stat-mech model easier to visualize. This property and the fact that the brickwork architecture is universal for MBQC constitute the primary reasons we studied this architecture in the first place. Architectures with larger depth lead to stat-mech models that cannot be straightforwardly collapsed onto a single plane while maintaining the two-body nature of the interactions.

2. Simplifications when k = 2

As in previous examples, we examine the $k = 2$ case. In this case, we might as well decimate all the degree-2 nodes in the graph in Fig. 11(a). This yields a graph with entirely degree-3 nodes, as shown in Fig. 11(b). The graph has two kinds of links, both carrying standard Ising interactions. The vertical blue links have weights given by

$$\text{weight}(\langle s_u s_{u'} \rangle) = \begin{cases} q^2(q^2+1) & \text{if } \sigma_u \sigma_{u'} = e, \\ q^2(2q) & \text{if } \sigma_u \sigma_{u'} = (12), \end{cases} \quad (32)$$

while the horizontal light green links have weights given by

$$\text{weight}(\langle s_u s_{u'} \rangle)$$
$$= \begin{cases} \frac{1}{q^2(q^4-1)^2}(q^6+q^4-4q^3+q^2+1) & \text{if } \sigma_u \sigma_{u'} = e, \\ \frac{1}{q^2(q^4-1)^2}(2q^5-2q^4-2q^2+2q) & \text{if } \sigma_u \sigma_{u'} = (12). \end{cases}$$
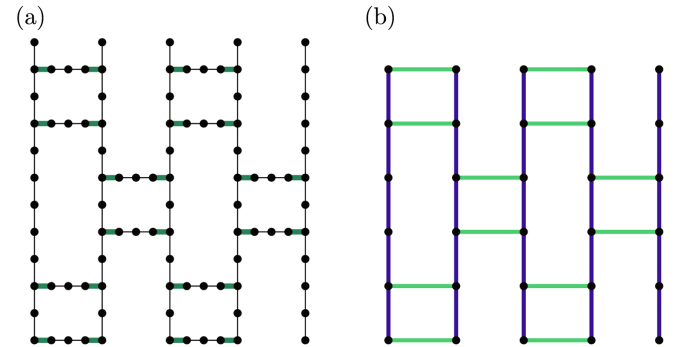$$(33)$$

FIG. 11. (a) The graph that results from decimating degree-1 nodes in Fig. 10(b). Each thin black link carries weight equal to the function $q^{C(\sigma\tau^{-1})}$, while each thick green link carries weight equal to $wg(\sigma\tau^{-1}, q^2)$. (b) The graph that results from decimating nodes of the graph in (a). For $k = 2$, both the horizontal light green and the vertical blue links are ferromagnetic but have different strengths.

Both of these interactions are ferromagnetic and become stronger as q increases. We may think of the model as the square lattice Ising model for which $1/2$ of the links carry a ferromagnetic interaction of one strength, $1/4$ of the links carry ferromagnetic interactions of another strength, and the final $1/4$ of the links have no interaction at all. The energy functional can be written

$$E/(kT) = -J_{\text{vert}} \sum_{\langle ij \rangle} s_i s_j - J_{\text{horiz}} \sum_{\langle ij \rangle} s_i s_j, \qquad (34)$$

where s_i take on values in $\{+1, -1\}$. For $q = 2$, we have $J_{\text{vert}} = \log(5/4)/2 = 0.112$ and $J_{\text{horiz}} = \log(53/28)/2 = 0.319$. Both of these values are weaker than the critical interaction strength for the square lattice Ising model of $J_{\text{square}} = \log(1 + \sqrt{2})/2 = 0.441$. This indicates that the graph generated by the stat-mech mapping on 2D circuits of depth 3 with brickwork architecture is in the disordered phase when $q = 2$. This remains true for $q = 3$. For $q = 4$, $J_{\text{horiz}} = 0.500 > J_{\text{square}}$, but $J_{\text{vert}} = 0.377 < J_{\text{square}}$. Recall that $1/4$ of the links can be thought to have $J = 0$, since they are missing. Taking this into account, the value of J averaged over all the links remains below J_{square} for $q = 5$ and slightly exceeds it for $q = 6$.

This indicates that when we run SEBD on these uniform depth-3 circuits with Haar-random gates, the quasientropy satisfies $\tilde{S}_2 = O(1)$ (independent of the number of qudits n) when $q = 2$ or $q = 3$ (and probably also for $q = 4$ and $q = 5$). We take this as evidence that SEBD would be efficient for these circuits.

VII. FUTURE WORK AND OPEN QUESTIONS

Our work yields several natural follow-up questions and places for potential future work. We list some here.

(1) Can ideas from our work also be used to simulate *noisy* 2D quantum circuits? Roughly, we expect that increasing noise in the circuit corresponds to decreasing the interaction strength in the corresponding stat-mech model, pushing the model closer toward the disordered phase, which is (heuristically) associated with efficiency of our algorithms. We therefore suspect that if noise is incorporated, there will be a three-dimensional phase diagram depending on circuit depth, qudit dimension, and noise strength. As the noise is increased, our algorithms may therefore be able to simulate larger depths and qudit dimensions than in the noiseless case.

(2) Can one approximately simulate random 2D circuits of arbitrary depth? This is the relevant case for Google's quantum computational supremacy experiment [2]. Assuming Conjecture 2, our algorithms are not efficient once the depth exceeds some constant, but it is not clear if this difference in apparent complexity for shallow vs deep circuits is simply an artifact of our simulation method, or if it is inherent to the problem itself.

(3) Our algorithms are well defined for all 2D circuits, not only random 2D circuits. Are they also efficient for other kinds of unitary evolution at shallow depths, for example, evolution by a fixed local 2D Hamiltonian for a short amount of time?

(4) Can we rigorously prove Conjecture 1? One way to make progress on this goal would be to find a worst-case-hard uniform circuit family for which it would be possible to perform the analytic continuation of quasientropies $\tilde{S}_k$ in the $k \to 1$ limit using the mapping to stat-mech models.

(5) Can we give numerical evidence for Conjecture 2, which claims that our algorithms undergo computational phase transitions? This would require numerically simulating our algorithms for circuit families with increasing local Hilbert space dimension and increasing depth and finding evidence that the algorithms eventually become inefficient.

(6) How precisely does the stat-mech mapping inform the efficiency of our algorithms? Is the correlation length of the stat-mech model associated with the run-time of our simulation algorithms? How well does the phase transition point in the stat-mech model (and accompanying phase transition in quasientropies) predict the computational phase transition point in the simulation algorithms? If such questions are answered, it may be possible to predict the efficiency and run-time of the simulation algorithms for an arbitrary (and possibly noisy) random circuit distribution via Monte Carlo studies of the associated stat-mech model. In this way, the performance of the algorithms could be studied even when direct numerical simulation is not feasible.

(7) In the regime where SEBD is inefficient, i.e., when the effective 1D dynamics it simulates are on the volume-law side of the entanglement phase transition, is SEBD still better than previously known exponential-time methods? Intuitively, we expect this to be the case close to the transition point.

(8) Can SEBD and/or Patching be generalized to simulate shallow circuits in three or higher dimensions? For SEBD the natural approach would be to use projected entangled pair states (higher-dimensional generalization of MPS) and simulate action of unitary gates and measurements, but projected entangled pair states cannot be efficiently contracted or truncated exactly in the same way as MPS.

The code for our implementation is available [106].

[1] A. W. Harrow and A. Montanaro, *Quantum Computational Supremacy*, Nature (London) **549**, 203 (2017).

[2] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, *Quantum Supremacy Using a Programmable Superconducting Processor*, Nature (London) **574**, 505 (2019).

[3] H.-S. Zhong *et al.*, *Quantum Computational Advantage Using Photons*, Science **370**, 1460 (2020).

[4] Y. Wu *et al.*, *Strong Quantum Computational Advantage Using a Superconducting Quantum Processor*, Phys. Rev. Lett. **127**, 180501 (2021).

[5] B. M. Terhal and D. P. DiVincenzo, *Adaptive Quantum Computation, Constant Depth Quantum Circuits and Arthur-Merlin Games*, Quantum Inf. Comput. **4**, 134 (2004).

[6] S. Aaronson and A. Arkhipov, *The Computational Complexity of Linear Optics*, in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11 (Association for Computing Machinery, New York, 2011), pp. 333–342.

[7] D. Shepherd and M. J. Bremner, *Temporally Unstructured Quantum Computation*, Proc. R. Soc. A **465**, 1413 (2009).

[8] M. J. Bremner, R. Jozsa, and D. J. Shepherd, *Classical Simulation of Commuting Quantum Computations Implies Collapse of the Polynomial Hierarchy*, Proc. R. Soc. A **467**, 459 (2011).

[9] M. J. Bremner, A. Montanaro, and D. J. Shepherd, *Average-Case Complexity versus Approximate Simulation of Commuting Quantum Computations*, Phys. Rev. Lett. **117**, 080501 (2016).

[10] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, *Characterizing Quantum Supremacy in Near-Term Devices*, Nat. Phys. **14**, 595 (2018).

[11] S. Aaronson and L. Chen, in *Proceedings of the 32nd Computational Complexity Conference*, CCC '17 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Germany, 2017), pp. 22:1–22:67.

[12] S. Aaronson, *Scott's Supreme Quantum Supremacy FAQ!—shtetl-optimized: The Blog of Scott Aaronson* [online; accessed 1 April 2021].

[13] I. L. Markov and Y. Shi, *Simulating Quantum Computation by Contracting Tensor Networks*, SIAM J. Comput. **38**, 963 (2008).

[14] D. Gottesman, *The Heisenberg Representation of Quantum Computers*, in *Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics, 1998* (International Press, Cambridge, MA, 1998), pp. 32–43.

[15] G. Vidal, *Efficient Classical Simulation of Slightly Entangled Quantum Computations*, Phys. Rev. Lett. **91**, 147902 (2003).

[16] G. Vidal, *Efficient Simulation of One-Dimensional Quantum Many-Body Systems*, Phys. Rev. Lett. **93**, 040502 (2004).

[17] T. J. Osborne, *Efficient Approximation of the Dynamics of One-Dimensional Quantum Spin Systems*, Phys. Rev. Lett. **97**, 157202 (2006).

[18] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, *Validating Quantum Computers Using Randomized Model Circuits*, Phys. Rev. A **100**, 032328 (2019).

[19] Our calculation of quantum volume for 2D circuits above uses the additional fact that, assuming for simplicity that $L_1 \leq L_2$, we can simulate a fully connected layer of gates on $L_2 x$ qubits (for $x \leq L_1$) with $O(x L_2 / L_1)$ locally connected 2D layers using the methods of Ref. [20]. Then, x is chosen to maximize $\min[L_2 x, d/(x L_2/L_1)]$.

[20] D. J. Rosenbaum, *Optimal Quantum Circuits for Nearest-Neighbor Architectures*, in *Proceedings of the 8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)* (Dagstuhl Publishing, Saarbrücken/Wadern, 2013), Vol. 22, pp. 294–307.

[21] C. Neill *et al.*, *A Blueprint for Demonstrating Quantum Supremacy with Superconducting Qubits*, Science **360**, 195 (2018).

[22] Y. Li, X. Chen, and M. P. A. Fisher, *Quantum Zeno Effect and the Many-Body Entanglement Transition*, Phys. Rev. B **98**, 205136 (2018).

[23] A. Chan, R. M. Nandkishore, M. Pretko, and G. Smith, *Unitary-Projective Entanglement Dynamics*, Phys. Rev. B **99**, 224307 (2019).

[24] B. Skinner, J. Ruhman, and A. Nahum, *Measurement-Induced Phase Transitions in the Dynamics of Entanglement*, Phys. Rev. X **9**, 031009 (2019).

[25] Y. Li, X. Chen, and M. P. A. Fisher, *Measurement-Driven Entanglement Transition in Hybrid Quantum Circuits*, Phys. Rev. B **100**, 134306 (2019).

[26] M. Szyniszewski, A. Romito, and H. Schomerus, *Entanglement Transition from Variable-Strength Weak Measurements*, Phys. Rev. B **100**, 064204 (2019).

[27] S. Choi, Y. Bao, X.-L. Qi, and E. Altman, *Quantum Error Correction in Scrambling Dynamics and Measurement-Induced Phase Transition*, Phys. Rev. Lett. **125**, 030505 (2020).

[28] M. J. Gullans and D. A. Huse, *Dynamical Purification Phase Transition Induced by Quantum Measurements*, Phys. Rev. X **10**, 041020 (2020).

[29] Y. Bao, S. Choi, and E. Altman, *Theory of the Phase Transition in Random Unitary Circuits with Measurements*, Phys. Rev. B **101**, 104301 (2020).

[30] C.-M. Jian, Y.-Z. You, R. Vasseur, and A. W. W. Ludwig, *Measurement-Induced Criticality in Random Quantum Circuits*, Phys. Rev. B **101,** 104302 (2020).

[31] M. J. Gullans and D. A. Huse, *Scalable Probes of Measurement-Induced Criticality*, Phys. Rev. Lett. **125,** 070606 (2020).

[32] A. Zabalo, M. J. Gullans, J. H. Wilson, S. Gopalakrishnan, D. A. Huse, and J. H. Pixley, *Critical Properties of the Measurement-Induced Transition in Random Quantum Circuits*, Phys. Rev. B **101,** 060301(R) (2020).

[33] Q. Tang and W. Zhu, *Measurement-Induced Phase Transition: A Case Study in the Nonintegrable Model by Density-Matrix Renormalization Group Calculations*, Phys. Rev. Research **2,** 013022 (2020).

[34] A. Nahum and B. Skinner, *Entanglement and Dynamics of Diffusion-Annihilation Processes with Majorana Defects*, Phys. Rev. Research **2,** 023288 (2020).

[35] K. Agarwal and N. Bao, *Toy Model for Decoherence in the Black Hole Information Problem*, Phys. Rev. D **102,** 086017 (2020).

[36] R. Fan, S. Vijay, A. Vishwanath, and Y.-Z. You, *Self-Organized Error Correction in Random Unitary Circuits with Measurement*, Phys. Rev. B **103,** 174309 (2021).

[37] Y. Li, X. Chen, A. W. W. Ludwig, and M. P. A. Fisher, *Conformal Invariance and Quantum Nonlocality in Hybrid Quantum Circuits*, Phys. Rev. B **104,** 104305 (2021).

[38] A. Lavasani, Y. Alavirad, and M. Barkeshli, *Measurement-Induced Topological Entanglement Transitions in Symmetric Random Quantum Circuits*, Nat. Phys. **17,** 342 (2021).

[39] S. Sang and T. H. Hsieh, *Measurement Protected Quantum Phases*, Phys. Rev. Research **3,** 023200 (2021).

[40] M. Ippoliti, M. J. Gullans, S. Gopalakrishnan, D. A. Huse, and V. Khemani, *Entanglement Phase Transitions in Measurement-Only Dynamics*, Phys. Rev. X **11,** 011030 (2021).

[41] Y. Fuji and Y. Ashida, *Measurement-Induced Quantum Criticality under Continuous Monitoring*, Phys. Rev. B **102,** 054302 (2020).

[42] M. Szyniszewski, A. Romito, and H. Schomerus, *Universality of Entanglement Transitions from Stroboscopic to Continuous Measurements*, Phys. Rev. Lett. **125,** 210602 (2020).

[43] S. Vijay, *Measurement-Driven Phase Transition within a Volume-Law Entangled Phase*, arXiv:2005.03052.

[44] O. Lunt and A. Pal, *Measurement-Induced Entanglement Transitions in Many-Body Localized Systems*, Phys. Rev. Research **2,** 043072 (2020).

[45] Y. Li and M. P. A. Fisher, *Statistical Mechanics of Quantum Error-Correcting Codes*, Phys. Rev. B **103,** 104306 (2021).

[46] X. Turkeshi, R. Fazio, and M. Dalmonte, *Measurement-Induced Criticality in $(2+1)$-Dimensional Hybrid Quantum Circuits*, Phys. Rev. B **102,** 014315 (2020).

[47] L. Fidkowski, J. Haah, and M. B. Hastings, *How Dynamical Quantum Memories Forget*, Quantum **5,** 382 (2021).

[48] A. Nahum, S. Roy, B. Skinner, and J. Ruhman, *Measurement and Entanglement Phase Transitions in All-to-All Quantum Circuits on Quantum Trees, and in Landau-Ginsburg Theory*, PRX Quantum **2,** 010352 (2021).

[49] M. Ippoliti and V. Khemani, *Postselection-Free Entanglement Dynamics via Spacetime Duality*, Phys. Rev. Lett. **126,** 060501 (2021).

[50] S. Aaronson, *Quantum Computing, Postselection, and Probabilistic Polynomial-Time*, Proc. R. Soc. A **461,** 3473 (2005).

[51] M. J. Bremner, A. Montanaro, and D. J. Shepherd, *Achieving Quantum Supremacy with Sparse and Noisy Commuting Quantum Computations*, Quantum **1,** 8 (2017).

[52] A. W. Harrow and S. Mehraban, *Approximate Unitary t-Designs by Short Random Quantum Circuits Using Nearest-Neighbor and Long-Range Gates*, arXiv:1809.06957.

[53] A. M. Dalzell, N. Hunter-Jones, and F. G. Brandão, *Random Quantum Circuits Anti-concentrate in Log Depth*, arXiv:2011.12277 [PRX Quantum (to be published)].

[54] A. Bouland, B. Fefferman, C. Nirkhe, and U. Vazirani, *On the Complexity and Verification of Quantum Random Circuit Sampling*, Nat. Phys. **15,** 159 (2019).

[55] R. Movassagh, *Quantum Supremacy and Random Circuits*, arXiv:1909.06210.

[56] A. Bouland, B. Fefferman, Z. Landau, and Y. Liu, *Noise and the Frontier of Quantum Supremacy*, arXiv:2102.01738.

[57] Y. Kondo, R. Mori, and R. Movassagh, *Fine-Grained Analysis and Improved Robustness of Quantum Supremacy for Random Circuit Sampling*, arXiv:2102.01960.

[58] D. N. Page, *Average Entropy of a Subsystem*, Phys. Rev. Lett. **71,** 1291 (1993).

[59] P. Hayden, D. W. Leung, and A. Winter, *Aspects of Generic Entanglement*, Commun. Math. Phys. **265,** 95 (2006).

[60] O. C. Dahlsten, R. Oliveira, and M. B. Plenio, *The Emergence of Typical Entanglement in Two-Party Random Processes*, J. Phys. A **40,** 8081 (2007).

[61] H. Liu and S. J. Suh, *Entanglement Tsunami: Universal Scaling in Holographic Thermalization*, Phys. Rev. Lett. **112,** 011601 (2014).

[62] B. Bertini, P. Kos, and T. c. v. Prosen, *Entanglement Spreading in a Minimal Model of Maximal Many-Body Quantum Chaos*, Phys. Rev. X **9,** 021033 (2019).

[63] P. Hayden, S. Nezami, X.-L. Qi, N. Thomas, M. Walter, and Z. Yang, *Holographic Duality from Random Tensor Networks*, J. High Energy Phys. 11 (2016) 009.

[64] M. B. Hastings, *The Asymptotics of Quantum Max-Flow Min-Cut*, Commun. Math. Phys. **351,** 387 (2017).

[65] M.-H. Yung and X. Gao, *Can Chaotic Quantum Circuits Maintain Quantum Supremacy under Noise?*, arXiv:1706.08913.

[66] X. Gao and L. Duan, *Efficient Classical Simulation of Noisy Quantum Computation*, arXiv:1810.03176.

[67] G. Kalai and G. Kindler, *Gaussian Noise Sensitivity and Boson Sampling*, arXiv:1409.3093.

[68] M. Oszmaniec and D. J. Brod, *Classical Simulation of Photonic Linear Optics with Lost Particles*, New J. Phys. **20,** 092002 (2018).

[69] L. Eldar and S. Mehraban, in *Proceedings of the 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, New York, 2018), pp. 23–34.

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

[71] S. Bravyi, D. Gosset, and R. Movassagh, *Classical Algorithms for Quantum Mean Values*, Nat. Phys. **17**, 337 (2021).

[72] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevX.12.021021 for deferred proofs, additional study of our simulation algorithm, additional details and analysis of the stat-mech mapping, and a proposal and analysis of an alternative simulation algorithm.

[73] P. W. Shor, in *Proceedings of the 37th Conference on Foundations of Computer Science* (IEEE, New York, 1996), pp. 56–65.

[74] D. Aharonov and M. Ben-Or, in *Proceedings of the 37th Conference on Foundations of Computer Science* (IEEE, New York, 1996), pp. 46–55.

[75] A. W. Harrow and M. A. Nielsen, *Robustness of Quantum Gates in the Presence of Noise*, Phys. Rev. A **68**, 012308 (2003).

[76] A. A. Razborov, *An Upper Bound on the Threshold Quantum Decoherence Rate*, Quantum Inf. Comput. **4**, 222 (2004).

[77] S. Virmani, S. F. Huelga, and M. B. Plenio, *Classical Simulability, Entanglement Breaking, and Quantum Computation Thresholds*, Phys. Rev. A **71**, 042328 (2005).

[78] H. Buhrman, R. Cleve, M. Laurent, N. Linden, A. Schrijver, and F. Unger, in *Proceedings of the 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)* (IEEE, New York, 2006), pp. 411–419.

[79] J. Kempe, O. Regev, F. Unger, and R. De Wolf, in *Proceedings of the International Colloquium on Automata, Languages, and Programming* (Springer, New York, 2008), pp. 845–856.

[80] R. Raussendorf, S. Bravyi, and J. Harrington, *Long-Range Quantum Entanglement in Noisy Cluster States*, Phys. Rev. A **71**, 062313 (2005).

[81] S. D. Barrett, S. D. Bartlett, A. C. Doherty, D. Jennings, and T. Rudolph, *Transitions in the Computational Power of Thermal States for Measurement-Based Quantum Computation*, Phys. Rev. A **80**, 062328 (2009).

[82] D. E. Browne, M. B. Elliott, S. T. Flammia, S. T. Merkel, A. Miyake, and A. J. Short, *Phase Transition of Computational Power in the Resource States for One-Way Quantum Computation*, New J. Phys. **10**, 023010 (2008).

[83] A. Deshpande, B. Fefferman, M. C. Tran, M. Foss-Feig, and A. V. Gorshkov, *Dynamical Phase Transitions in Sampling Complexity*, Phys. Rev. Lett. **121**, 030501 (2018).

[84] G. Muraleedharan, A. Miyake, and I. H. Deutsch, *Quantum Computational Supremacy in the Sampling of Bosonic Random Walkers on a One-Dimensional Lattice*, New J. Phys. **21**, 055003 (2019).

[85] R. Raussendorf and H. J. Briegel, *A One-Way Quantum Computer*, Phys. Rev. Lett. **86**, 5188 (2001).

[86] X. Gao, S.-T. Wang, and L.-M. Duan, *Quantum Supremacy for Simulating a Translation-Invariant Ising Spin Model*, Phys. Rev. Lett. **118**, 040502 (2017).

[87] J. Miller, S. Sanders, and A. Miyake, *Quantum Supremacy in Constant-Time Measurement-Based Computation: A Unified Architecture for Sampling and Verification*, Phys. Rev. A **96**, 062320 (2017).

[88] J. Bermejo-Vega, D. Hangleiter, M. Schwarz, R. Raussendorf, and J. Eisert, *Architectures for Quantum Simulation Showing a Quantum Speedup*, Phys. Rev. X **8**, 021010 (2018).

[89] Unless specified otherwise, we use *worst-case simulation* to refer to the problem of exactly simulating an arbitrary circuit instance.

[90] A. Broadbent, J. Fitzsimons, and E. Kashefi, in *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, New York, 2009), pp. 517–526.

[91] I. H. Kim, *Holographic Quantum Simulation*, arXiv:1702.02093.

[92] I. H. Kim, *Noise-Resilient Preparation of Quantum Many-Body Ground States*, arXiv:1703.00032.

[93] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, *Entropy Scaling and Simulability by Matrix Product States*, Phys. Rev. Lett. **100**, 030504 (2008).

[94] R. Orús, *A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States*, Ann. Phys. (Amsterdam) **349**, 117 (2014).

[95] J. C. Bridgeman and C. T. Chubb, *Hand-Waving and Interpretive Dance: An Introductory Course on Tensor Networks*, J. Phys. A **50**, 223001 (2017).

[96] M. Horodecki, J. Oppenheim, and A. Winter, *Quantum State Merging and Negative Information*, Commun. Math. Phys. **269**, 107 (2006).

[97] B. Villalonga, D. Lyakh, S. Boixo, H. Neven, T. S. Humble, R. Biswas, E. G. Rieffel, A. Ho, and S. Mandrà, *Establishing the Quantum Supremacy Frontier with a 281 pflop/s Simulation*, Quantum Sci. Technol. **5**, 034003 (2020).

[98] A. Nahum, S. Vijay, and J. Haah, *Operator Spreading in Random Unitary Circuits*, Phys. Rev. X **8**, 021014 (2018).

[99] C. W. von Keyserlingk, T. Rakovszky, F. Pollmann, and S. L. Sondhi, *Operator Hydrodynamics, OTOCs, and Entanglement Growth in Systems without Conservation Laws*, Phys. Rev. X **8**, 021013 (2018).

[100] T. Zhou and A. Nahum, *Emergent Statistical Mechanics of Entanglement in Random Unitary Circuits*, Phys. Rev. B **99**, 174205 (2019).

[101] N. Hunter-Jones, *Unitary Designs from Statistical Mechanics in Random Quantum Circuits*, arXiv:1905.12053.

[102] A bound on S_k constrains the tail of the spectrum of singular values $\{\sigma_j\}$ of a state ρ, where σ_j appear in nonincreasing order. For bond dimension D, let $\epsilon \coloneqq \sum_{j=D+1}^{\infty} \sigma_j$ be the associated truncation error. Note that $\epsilon \leq \Pr \sigma_x \leq 1/D$, where x is a random variable that takes value j with probability σ_j. Then, for any $k < 1$, using Markov's inequality on the random variable σ_x^{k-1}, we find

$$\epsilon \leq \Pr\{\sigma_x^{k-1} \geq (1/D)^{k-1}\} \leq \mathbb{E}[\sigma_x^{k-1}]D^{k-1} = (e^{S_k}/D)^{1-k},$$

and, thus, $D \geq e^{S_k}\epsilon^{-1/(1-k)}$ is sufficient to guarantee ϵ truncation error at bond dimension D. In particular, when $S_k = O[\log(n)]$, $D = \text{poly}(n)$ ensures $\epsilon \leq 1/\text{poly}(n)$. On the other hand, for $k = 1$, there exist counterexamples [93] that show a bound on S_1 cannot alone be sufficient for SEBD's efficiency.

[103] Y. Huang, *Dynamics of Renyi Entanglement Entropy in Local Quantum Circuits with Charge Conservation*, IOP SciNotes **1**, 035205 (2020).

[104] R. Vasseur, A. C. Potter, Y.-Z. You, and A. W. W. Ludwig, *Entanglement Transitions from Holographic Random Tensor Networks*, Phys. Rev. B **100**, 134203 (2019).

[105] Y. Gu, *Moments of Random Matrices and Weingarten Functions*, Ph.D. thesis, Queen's University, 2013.

[106] https://github.com/random-shallow-2d/random-shallow-2d.