Supplemental Material for "Sharp complexity phase transitions generated by entanglement"

Soumik Ghosh, Abhinav Deshpande, Dominik Hangleiter, Alexey V. Gorshkov, and Bill Fefferman

CONTENTS

S1. SETTING

In this section, we define our setup and define some conventions that we use in this work. For a graph G, denote the corresponding graph state as $|G\rangle$. We consider graph states on k–regular graphs for $k \in [n-1] \equiv \{1, 2, \ldots, n-1\}$.

Definition 4 (k–regular graph)**.** *A k-regular graph is a graph in which the degree of every vertex, that is, the number of adjacent vertices, is exactly k.*

We then consider measurements of the qubits, or in other words, the vertices of G in an arbitrary single-qubit basis. Such a measurement is equivalent to the application of a product of arbitrary single qubit gates followed by a standard basis measurement.

For each qubit $i \in [n] \cong V$, we parameterize the single-qubit unitary U_i as

$$U_i(\theta_i, \phi_i) = \begin{pmatrix} \cos\frac{\theta_i}{2} & -\sin\frac{\theta_i}{2} \\ e^{i\phi_i}\sin\frac{\theta_i}{2} & e^{i\phi_i}\cos\frac{\theta_i}{2} \end{pmatrix}. \tag{1}$$

When the arguments of $U_i(\theta_i, \phi_i)$ are clear from the context, we drop them and just use U_i. Let

$$\mathsf{U} = U_1 \otimes U_2 \otimes \cdots \otimes U_n. \tag{2}$$

For fixed G and U, the probability of getting any outcome $x \in \{0,1\}^n$ is given by

$$p_x(G, \mathsf{U}) = \left| \langle x | \bigotimes_{i=1}^{n} U_i |G\rangle \right|^2. \tag{3}$$

A closely related quantity, the *probability amplitude*, is defined as

$$q_x(G, \mathsf{U}) = \langle x | \bigotimes_{i=1}^{n} U_i |G\rangle. \tag{4}$$

When the context is clear from usage, we drop either G and U or both from our notation. Additionally, let us denote by $\mathcal{D}(G, \mathsf{U})$ the following probability distribution:

$$\Pr_{X \sim \mathcal{D}(G, U^{(l)})}[X = x] = p_x(G, \mathsf{U}). \tag{5}$$

This probability distribution is over x, for a fixed G and U.

Equivalently, we can view our setting in the standard circuit picture. In this picture, the family of quantum circuits we are going to study contain a Clifford part, which is used to construct the graph state $|G\rangle$. In this part, there is a first layer of Hadamard gates, followed by a sequence of controlled-Z gates applied on the edges of G. This is followed by the layer of single-qubit gates and a standard basis measurement, as illustrated in Figure 1 of the main text.

Throughout this work, we let the symbols X, Y, and Z denote single qubit Pauli-X, Pauli-Y, and Pauli-Z gates respectively. Additionally, we use capital letters, like U, V et cetera, to denote quantum gates.

S2. TECHNIQUES

In this section, we talk about manipulating graph states and also talk about some useful complexity theoretic results that we use in this work. First, we start

with basic concepts in graph theory and how they relate to graph states. Then, we discuss how we measure entanglement in the rest of the work. Finally, we introduce some advanced concepts in graph theory and use them to find upper and lower bounds on our entanglement measure.

A. Some basic concepts in graph theory

Let us begin by briefly rehashing some well-known concepts from graph theory.

Consider a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. All the graphs, unless otherwise stated, are simple and undirected. Sometimes, to avoid ambiguity, we use V_G and E_G to denote that the vertex and edge sets of G. We state a series of standard definitions from graph theory below. We assume some familiarity with other related notions from graph theory, which can be found in [62].

Definition 5 (Adjacency matrix)**.** *The* adjacency matrix *of a graph G is a $|V| \times |V|$ matrix A such that*

$$\begin{aligned} A_{i,j} &= 1 \quad \text{if } (u,v) \in E, \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{6}$$

Definition 6 (Neighborhood)**.** *The* neighborhood *of a vertex v in the graph G is defined as*

$$N_v(G) = \{u : (u,v) \in E\}. \tag{7}$$

For two graphs G and H such that H is a subgraph of G, the operation $K = G \setminus H$ is defined as

$$K = (V_G,\ E_G \setminus E_H). \tag{8}$$

Definition 7 (Complete graph)**.** *A* complete graph *is a graph in which every pair of distinct vertices is connected by an edge.*

Definition 8 (Complement of a graph)**.** *The* complement *of a graph G with n vertices, denoted by $\overline{G}$, is defined as*

$$\overline{G} = G_{complete} \setminus G, \tag{9}$$

where $G_{complete}$ is the complete graph on n vertices.

Definition 9 (Subgraph)**.** *A* subgraph *of a graph $G = (V, E)$ is another graph whose vertex and edge sets are subsets of V and E respectively.*

Definition 10 (Tree)**.** *A* tree *is a connected, acyclic graph.*

Definition 11 (Binary tree)**.** *A* binary tree *is a tree where each vertex is connected to at most three other vertices.*

Definition 12 (Bipartite graph)**.** *A* bipartite graph *is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V.*

Definition 13 (Complete bipartite graph)**.** *A* complete bipartite graph *is a bipartite graph whose vertices can be divided into two disjoint sets U and V such that*

(a) every edge connects a vertex in U to one in V, and

(b) every vertex of U is connected to every vertex of V.

A complete bipartite graph is denoted by $K_{p\times q}$, where p and q are the cardinalities of the sets U and V, respectively.

B. Operations on graph states

We frequently make use of a graph-theoretic operation called *vertex deletion*. We define it as follows.

Definition 14 (Vertex deletion)**.** *For a graph $G = (V, E)$, the operation* vertex deletion*, when applied to the vertex v of G, produces a new graph H such that*

$$H = (V \setminus v, E \setminus S), \tag{10}$$

where S is the set of all the edges incident to v.

Applying vertex deletion to a vertex v of G is equivalent to measuring the corresponding qubit of $|G\rangle$ in the Z basis. Indeed, let $P_v^{(\mathsf{Z},\pm)}$ be the two Pauli projectors onto the Z basis when we measure vertex v. Let $N(v)$ be the neighborhood of v and let us delete the vertex v to get the graph H. Then,

$$\begin{aligned} P_v^{(\mathsf{Z},+)} |G\rangle &= \frac{1}{\sqrt{2}} |0\rangle |H\rangle, \\ P_v^{(\mathsf{Z},-)} |G\rangle &= \frac{1}{\sqrt{2}} |1\rangle \prod_{u\in N_v(G)} Z_u |H\rangle. \end{aligned} \tag{11}$$

Note that the gates Z_u in equation (11) are single-qubit gates. Hence, by an appropriate choice of a last layer of local rotations, we can implement a vertex deletion.

Before defining the next graph operation, let us define the symmetric difference operator between two sets.

Definition 15 (Symmetric difference)**.** *The symmetric difference operator between two sets A and B, denoted by Δ, is defined as*

$$A \,\Delta\, B = (B \setminus A) \cup (A \setminus B). \tag{12}$$

Definition 16 (Local complementation)**.** *For a graph $G = (V, E)$, a* local complementation *τ_v on the vertex v flips the neighborhood of v: two vertices which were previously connected (in the neighborhood of v) get disconnected and two vertices which were previously disconnected get connected. The neighborhood of a vertex u in the graph $\tau_v(G)$ is given by*

$$\begin{aligned} N_u(\tau_v(G)) &= N_u(G) \,\Delta\, (N_v \setminus u) \quad \text{if } (u,v) \in E, \\ &= N_u(G) \quad \text{otherwise.} \end{aligned} \tag{13}$$

It is known that applying a local complementation to a vertex v in G is equivalent to applying a sequence of single qubit Clifford unitaries

$$U = \exp\left(-i\frac{\pi}{4}X_v\right)\prod_{u\in N_v}\exp\left(i\frac{\pi}{4}Z_u\right) \tag{14}$$

to the graph state $|G\rangle$. That is,

$$|\tau_v(G)\rangle = U_v\,|G\rangle\,. \tag{15}$$

Local complementation is illustrated in Figure 4 of the main text.

C. Some useful results from complexity theory

Before stating our technical lemmas, we state a few results from complexity theory for convenience. Variants of these statements and proofs have appeared in [6, 7, 38, 70, 71], so we only state the lemmas without proofs.

Lemma 17. *Consider an n–vertex graph R_n such that the corresponding graph state $|R_n\rangle$ is a resource state for measurement based quantum computation. Consider local rotations* U*, as defined in equation (2). Then, computing $p_x(R,\mathsf{U})$ is* GapP*-hard in the worst case over* U*, for any $x\in\{0,1\}^n$, up to constant multiplicative error.*

The proof follows from [6, 24, 47, 48]. For example, considering graphs with n vertices, this fact holds for an $\sqrt{n}\times\sqrt{n}$ grid graph. The idea is that resource states can do universal quantum computing under post-selection. By standard arguments sketched in these papers, that implies computing probabilities is GapP-hard.

Lemma 18. *Let R be an n–vertex graph such that $|R_n\rangle$ is a resource state for measurement based quantum computation. Let T be a graph such that R can be reached from T by vertex deletion and local complementation. Then, computing $p_x(T,\mathsf{U})$ is* GapP*-hard in the worst case over* U*, for every $x\in\{0,1\}^n$, up to constant multiplicative error.*

The proof follows from [47, 50, 72]. The intuition is that both vertex deletion and local complementation correspond to single-qubit operations, as we saw in Section S2 B. Hence, R is equivalent to a locally rotated grid graph, which is a known resource state.

Lemma 19 ([37])**.** *Let C be a quantum circuit and let*

$$p_x = |\langle x|C|0^n\rangle|^2, \tag{16}$$

for $x\in\{0,1\}^n$. Consider a distribution $\mathcal{D}$ given by

$$\Pr_{X\sim\mathcal{D}}[X=x] = p_x. \tag{17}$$

Note that this is a distribution over x for a fixed C. Then, if there is a polynomial time classical algorithm to sample from $\mathcal{D}$, then there is a $\mathsf{BPP}^{\mathsf{NP}}$ *algorithm to estimate p_x, upto constant multiplicative error, for a random choice of $x\in\{0,1\}^n$.*

Consequently, if computing every p_x is GapP*-hard upto constant multiplicative precision, no polynomial time classical algorithm exists to sample from $\mathcal{D}$, assuming the* PH *does not collapse to* $\mathsf{BPP}^{\mathsf{NP}}$*.*

The proof follows from Stockmeyer's counting theorem [37] and can also be found in [38]. Lemma 19 states that assuming the PH does not collapse to $\mathsf{BPP}^{\mathsf{NP}}$, one can rule out the existence of classical exact samplers which sample from the output distribution of C. Through additional appropriate conjectures about the additive error hardness of computing the output probabilities of C, one can also extend the result of Lemma 19 to rule out classical approximate samplers, up to an appropriate error in total variation distance, using standard techniques which are also outlined in [38, 73, 74]. Qualitatively, the proof says that the two popular characterizations of simulation of quantum circuits—sampling and estimation—are interlinked. The presence of a sampler in BPP implies the presence of an estimator in $\mathsf{BPP}^{\mathsf{NP}}$.

D. Measuring entanglement entropy

Let $|\psi\rangle$ be an n-qubit pure state and let (A,B) be a partition of the qubits and let the corresponding Hilbert spaces be $\mathcal{H}_\mathsf{A}$ and $\mathcal{H}_\mathsf{B}$. Consider a Schmidt decomposition of $|\psi\rangle$ as follows:

$$|\psi\rangle = \sum_{i=1}^{\min\left(2^{|\mathsf{A}|},2^{|\mathsf{B}|}\right)}\sqrt{s_i}\,|u_i\rangle\,|v_i\rangle\,, \tag{18}$$

where $\left\{|u_i\rangle : i\in\{1,2,\ldots,2^{\mathcal{H}_{|\mathsf{A}|}}\}\right\}$ and $\left\{|v_i\rangle : i\in\{1,2,\ldots,2^{\mathcal{H}_{|\mathsf{B}|}}\}\right\}$ are two sets of orthonormal basis vectors of the Hilbert spaces $\mathcal{H}_\mathsf{A}$ and $\mathcal{H}_\mathsf{B}$, respectively. Here are s_i are nonnegative real numbers. The Schmidt rank of a quantum state is given by the number of non-zero s_i-s in the Schmidt decomposition.

Then the von Neumann bipartite entanglement entropy of $|\psi\rangle$, with respect to the bipartition (A,B) is given by

$$S(|\psi\rangle_{\mathsf{A},\mathsf{B}}) = \sum_{i=1}^{\min\left(2^{|\mathsf{A}|},\,2^{|\mathsf{B}|}\right)} -s_i\log s_i. \tag{19}$$

There are other measures of entanglement, like the logarithm of the Schmidt rank of a quantum state, with respect to a bipartition. Schmidt rank width is a natural generalization of the Schmidt rank of a quantum state.

Definition 20 (Schmidt rank width)**.** *Consider an n-qubit state $|\psi\rangle$. Consider trees T with exactly n leaves where the*

maximum degree of each vertex is 3. *Any edge e in the tree splits the leaves of the tree into two sets A_e and B_e, depending on the two connected components of $T - e$, where $T - e$ is a graph with edge e removed. Then, the Schmidt rank width of $|\psi\rangle$ is given by*

$$\mathsf{srw}(|\psi\rangle) = \min_{T} \max_{e \in T} \log_2 \mathsf{r}_{\mathsf{A}_e,\mathsf{B}_e}, \tag{20}$$

where $r_{\mathsf{A}_e,\mathsf{B}_e}$ is the Schmidt rank with respect to the bipartition $\mathsf{r}_{\mathsf{A}_e,\mathsf{B}_e}$.

Qualitatively, it measures how large a linear combination we need, in the worst case, to write down the quantum state $|\psi\rangle$ for the best "tree-like" decomposition of the state. The more "succinct" and more "tree-like" the state is, the less is the resource overhead in simulating the state by tree tensor networks [11, 12, 17]. Therefore, Schmidt-rank width is a measure of the worst-case simulation cost, provided we can figure out an optimal tree decomposition.

1. Entanglement width

In this paper, we measure entanglement entropy in terms of *entanglement width*, defined in [25].

Definition 21 (Entanglement width)**.** *Consider an n-qubit state $|\psi\rangle$. Consider trees T with exactly n leaves where the maximum degree of each vertex is* 3. *Any edge e in the tree splits n qubits into two sets, A_e and B_e, depending on the two connected components of $T - e$, where $T - e$ is a graph with edge e removed. Then the entanglement width of $|\psi\rangle$ is given by*

$$\mathsf{ew}(|\psi\rangle) = \min_{T} \max_{e \in T} S\left(|\psi\rangle_{\mathsf{A}_e,\mathsf{B}_e}\right). \tag{21}$$

The qualitative interpretation of this measure is the same as that of the Schmidt rank width: it measures how entangled the state is across "tree-like" bipartitions.

E. Relation to classical simulations

Entanglement width is important because the runtime of tensor network simulations of a graph state depends on its entanglement width. To formalize this, let us state the following lemma.

Lemma 22 ([17])**.** *Let $|G\rangle$ be an n qubit graph state. Then, using tree-tensor networks, $p_x(G, \mathsf{U})$ can be exactly computed in poly $\left(n, 2^{\mathsf{ew}(|G\rangle)}\right)$ time, for any choice of U.*

Additionally, using tree-tensor networks, the output distribution $\mathcal{D}(G, \mathsf{U})$ can be sampled from in poly $\left(n, 2^{\mathsf{ew}(|G\rangle)}\right)$ time, for any choice of U.

An immediate consequence of Lemma 22 is that, if the entanglement width of a graph state is upper bounded by $\mathcal{O}(\log n)$, classical simulations are efficient. Also note that implicit in Lemma 22 is the fact that we can efficiently find the optimal tree decomposition for $|G\rangle$ in polynomial time, when the entanglement width is logarithmically bounded.

F. Width-measures in graph theory

Here, we define the notions of tree width, clique width, and rank width. We had merely referenced these measures in the main text: we look at them in detail below.

1. Tree width

First, we look at a measure called "tree width." The definitions are taken from [62]. Intuitively, tree width measures how "similar" a graph is to a tree. The tree width of any tree is 1.

Definition 23 (Tree decomposition)**.** *A* tree decomposition *of a graph $G = (V, E)$ is a tree T such that the following properties hold.*

(a) Each vertex i of T is labelled by a subset $B_i \subset V$. Each such vertex is called a "bag".

(b) The two vertices corresponding to every edge in E are both in at least one B_i.

(c) For every vertex $u \in V$, the subtree of T consisting of all the "bags" containing u is connected.

Definition 24. *The* width *of a tree decomposition T is given by $\max_i |B_i| - 1$.*

Definition 25 (Tree width)**.** *The* tree width *of a graph G, denoted by $\mathsf{tw}(G)$, is the minimum number t such that there exists a tree decomposition with width at most t.*

Many tasks that are NP-hard in general are efficiently solvable in graphs with bounded tree width [61]. Graph states with logarithmically bounded tree width can be efficiently simulated under arbitrary local rotations [11] if we know the corresponding tree decomposition.

Although the tree width is a very useful concept in these regards, it can be unbounded in "dense" graphs, i.e. graphs that have lots of edges. However, these graphs can otherwise have a lot of symmetry and many problems can still be easy to solve in these graphs. For example, the complete graph on n vertices has a tree width of $n - 1$, but a lot of problems are easy when restricted to just the complete graph. Furthermore, deciding whether the tree width of a graph is at most m for a given integer m is NP-complete [57].

With that motivation in mind, some generalizations of tree width become necessary.

2. Clique width

The definition is taken from [41].

Definition 26 (Clique width)**.** *The* clique width *of a graph G, denoted by $\mathsf{cw}(G)$, is defined as the minimum number of labels needed to construct G with the following operations:*

(a) Creating a new vertex v that is labeled by an integer i.

(b) If there are two graphs G and H which are already constructed, taking a disjoint union of these two graphs to create a new graph.

(c) Creating a new edge between all vertices of label i and all vertices of label j.

(d) Changing the label of a vertex.

Note that the clique width is a generalization of tree width, in the sense that graphs with bounded tree width also have bounded clique width. However, the clique width can remain bounded even when the tree width blows up. For example, the clique width of the complete graph is 2. In that sense, it is a "more useful" metric than the tree width. However, deciding whether the clique width of a graph is at most m for a given integer m is NP-complete [56]. So, practically, using this measure might be difficult.

We now define a generalization of clique width that does not have this problem.

3. Rank width

The definitions are taken from [42].

Definition 27 (Cut rank)**.** *Let M be the $|V| \times |V|$ adjacency matrix of the graph G. The* cut rank *of $A \subseteq V$ is the rank of the submatrix of M with row labels corresponding to A and column labels corresponding to $V \setminus A$.*

Definition 28 (Rank decomposition)**.** *A* rank decomposition *of $G = (V, E)$ is a pair (T, L) where T is a binary tree and L is a bijection from V to the leaves of the tree.*

For a particular rank decomposition (T, L) of a graph G, any edge e in the tree T splits V into two parts, A_e and B_e, corresponding to the two connected components of $T - e$, where $T - e$ is the tree T with edge e removed. The *width* of an edge e is the cut-rank of A_e (which is equivalent to the cut-rank of B_e).

Definition 29. *The* width of the rank decomposition (T, L) *is the maximum width of an edge in T.*

Definition 30 (Rank width)**.** *The* rank width *of a graph G, denoted by $\mathsf{rw}(G)$, is the minimum width of a rank decomposition of G.*

Rank width generalizes clique width, in the sense that graphs with bounded clique width also have bounded rank width. Additionally, there is a polynomial time algorithm to decide whether the rank width of a graph is at most k, for a given integer k [41].

4. Inter-relations between width measures

Here, we state a few inter-relations between width measures. Proofs can be found in [41, 42, 59].

First, we upper bound clique width in terms of tree width.

Lemma 31 ([54])**.** *For a graph G,*

$$\mathsf{cw}(G) \leq 3 \cdot 2^{\mathsf{tw}(G)-1}. \tag{22}$$

Next, we lower bound clique width in terms of the tree width.

Lemma 32 ([59])**.** *Let G be an n–vertex graph such that it does not have the complete bipartite graph $K_{t\times t}$ as a subgraph, for some value of t. Then,*

$$\mathsf{tw}(G) \leq 3 \cdot \mathsf{cw}(G) \cdot (t-1) - 1. \tag{23}$$

Now, we relate rank width and clique width.

Lemma 33 ([42])**.** *For a graph G,*

$$\mathsf{rw}(G) \leq \mathsf{cw}(G) \leq 2^{\mathsf{rw}(G)+1} - 1. \tag{24}$$

Now, we state a relation between the clique width of a graph and its complement.

Lemma 34 ([41])**.** *For a graph G,*

$$\frac{1}{2} \cdot \mathsf{cw}(G) \leq \mathsf{cw}(\overline{G}) \leq 2 \cdot \mathsf{cw}(G), \tag{25}$$

where $\overline{G}$ is the complement of G.

Finally, we state one more lemma about how operations like vertex deletion do not increase the clique width. This helps us in independently analyzing the clique width of some of our hard graphs.

Lemma 35 ([41])**.** *Let G be a graph and H be obtained from G by a sequence of vertex deletions. Then,*

$$\mathsf{cw}(H) \leq \mathsf{cw}(G). \tag{26}$$

G. Relation to entanglement width

Here, we relate the width measures we saw from graph theory to the entanglement measures we saw in the study of quantum states. We relate these two using graph states.

In general, for any state $|\psi\rangle$, since

$$S(|\psi\rangle_{\mathsf{A},\mathsf{B}}) \leq \log_2 (\mathsf{r}_{\mathsf{A},\mathsf{B}}) \tag{27}$$

for any bipartition (A, B), we have

$$\mathsf{ew}(|\psi\rangle) \leq \mathsf{srw}(|\psi\rangle). \tag{28}$$

However, the situation greatly simplifies for a graph state. We now state a lemma from [17] to show this.

Lemma 36. *For any graph state* $|G\rangle$,

$$\mathsf{ew}(|G\rangle) = \mathsf{srw}(|G\rangle) = \mathsf{rw}(G). \tag{29}$$

From Lemma 36, entanglement width has exactly the same physical interpretation as the Schmidt rank width for a graph state.

Additionally, by exploiting the connections between rank width and entanglement width, we can argue that entanglement width satisfies the following upper and lower bounds, which can be found in [42].

Lemma 37. *For a graph* G *and the corresponding graph state* $|G\rangle$,

$$\mathsf{ew}(|G\rangle) \leq \mathsf{cw}(G) \leq 2^{\mathsf{ew}(|G\rangle)+1} - 1. \tag{30}$$

S3. RESULTS

In this section, first we state our easiness and hardness results. Then, we give a formal proof for each of them, one after another. We also give a formal statement and proof of the duality theorem. A description of the proofs was already provided in the main text. Here, we provide more rigorous technical details, for the sake of completeness.

Finally, we use our hardness results to sharpen the bound on the entanglement width of certain graphs.

Let $\mathcal{G}_k$ be the set of all k–regular graphs. Consider the following task.

RegularGraph$[n, k, x, \mathsf{U}]$

Input A description of an n–vertex, k–regular graph $G \in \mathcal{G}_k$, $x \in \{0,1\}^n$, and a description of the last layer of local rotations U.

Output An inverse polynomial multiplicative error estimate of $p_x(G, \mathsf{U})$.

Theorem 38 (Easy cases)**.** *For* $k \in \{1, 2, n-3, n-2, n-1\}$, $\mathsf{ew}(|G\rangle)$ *is a constant for any* $G \in \mathcal{G}_k$ *and* RegularGraph$[n, k, x, \mathsf{U}]$ *is solvable in classical polynomial time.*

Theorem 39 (Hard cases)**.** *For* $3 \leq k \leq n-4$, RegularGraph$[n, k, x, \mathsf{U}]$ *is* GapP*-hard, for any* $x \in \{0,1\}^n$*, assuming the* PH *does not collapse to* $\mathsf{BPP}^{\mathsf{NP}}$.

We delegate the proof of these two theorems to Section S3 A and Section S3 D, respectively. Two corollaries are evident.

Corollary 40. *For* $3 \leq k \leq n-4$*, there exists an explicit, efficient construction of a family of* n*–vertex,* k*–regular graphs* $\mathcal{F}$ *such that*

$$\mathsf{ew}(|G\rangle) = \omega(\log n), \tag{31}$$

for any $G \in \mathcal{F}$*, assuming that the* PH *does not collapse to* $\mathsf{BPP}^{\mathsf{NP}}$.

Proof. Follows from Lemma 22 and Theorem 39. □

Note that, in our hardness results, we prove something more fine-grained than what is required for Theorem 39. We construct k–regular graphs for which it is GapP-hard to compute the output probabilities to inverse polynomial multiplicative precision, under local rotations, and then invoke Lemma 19 to prove Theorem 39.

Consequently, we talk about how we can have weaker complexity theoretic conjectures imply the same bound as that of equation (31) and how we can have sharper bounds by tightening the conjecture, in Section S3 E 2.

We also prove a duality between regimes of low regularity and regimes of high regularity, which serves as a convenient tool for proving our hardness results. The duality, as is later discussed, follows from Theorem 41.

Theorem 41 (Duality theorem)**.** *The complement of an* n*–vertex hexagonal lattice, or an* $n \times n$ *grid graph* G *is a resource state for measurement based quantum computing.*

We prove this for the grid graph, and observe that a similar result holds for the hexagonal lattice. We convert the complement of an $n \times n$ grid graph to an $(n-1) \times (n-1)$ grid graph, by a sequence of vertex deletions and local complementation. To prove hardness, we reduce some of our hard graphs to the complement of a grid graph.

A. Proof of Theorem 38 (The easy regime)

Let $\mathcal{G}_k$ be the set of all k–regular graphs. Note that $\mathsf{cw}(G) = 2$ when G is the complete graph [58]. Additionally, note that

$$\mathsf{tw}(G) = 1 \tag{32}$$

for every $G \in \mathcal{G}_1$. This is because an n vertex 1–regular graph is just $n/2$ disconnected lines, each of unit length—so, each can be thought of as a tree with two nodes and one edge and would have a treewidth of 1—and treewidth does not change by a disjoint union of graphs with same treewidth.

Also note that, for $k = 2$,

$$\mathsf{tw}(G) = 2 \tag{33}$$

for every $G \in \mathcal{G}_2$. This is because every 2–regular graph is a disjoint union of a number of series-parallel graphs[**?**], each of which has a tree width of 2. Series-parallel graphs draw their inspiration from series-parallel electrical circuits and consist of connected vertices either in series, or in parallel, or a combination of both.

Hence, by Lemma 33 and Lemma 31, the clique width and rank width of every graph in $\mathcal{G}_1$ and $\mathcal{G}_2$ is bounded by a constant.

By Lemma 34, the complement of a bounded clique width graph also has bounded clique width. Since every

$(n-2)$– and $(n-3)$–regular graph is a complement of some 1–regular or 2–regular graph, the clique width and the rank width of every $(n-1)$–regular and $(n-2)$–regular graph is also bounded by a constant. By Lemma 36, the entanglement width of any graph state $|G\rangle$ is exactly the rank width of the corresponding graph G. Hence, the entanglement width of every 1–, 2–, $(n-1)$–, $(n-2)$–, and $(n-3)$–regular graph is bounded.

B. The complete graph revisited

We had discussed the recursive method for computing the output probabilities of the complete graph in the main text. In this section, we elucidate the technical details behind the recursive method.

We have just established the that the entanglement width of the complete graph is bounded, and hence, by Lemma 19, efficient sampling and probability estimation is possible under any local rotation. Here, we give a different proof of the fact that output probabilities of the complete graph can be computed in polynomial time, under any local rotation. Our proof leverages inherent symmetry properties of the complete graph.

The easiness of computing output probabilities of the complete graph has been established before, by observing that a complete graph has bounded clique width [25], by observing that a complete graph is reducible to a star graph under local Clifford rotations and the fact that the star graph has bounded tree width [26], or by observing that the complete graph can be reduced by local Clifford rotations [7] to the GHZ state which can be efficiently classically simulated. But, to the best of our knowledge, our approach has not been taken before and may be utilized elsewhere where there is Hamming weight symmetry.

The motivation behind this new approach is that our result rely on symmetry properties of the complete graph: this proof can serve as a refresher of those properties. The one specific property we repeatedly use is the fact that the output probabilities of a complete graph has Hamming weight symmetry—it can be written as a linear combination of polynomially many terms, one for each Hamming weight, such that each of them is efficiently computable. In this easiness proof, we can actually go slightly beyond the setting of Figure 1 by allowing the use of $e^{i\theta Z_i Z_j}$ gates and not just controlled-Z gates. In more technical terms, define the following state $|G\rangle$ on a complete graph:

$$|G\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} \beta^{\left(\sum_{i,j\in[n],i<j} z_i z_j\right)} |z\rangle\,, \tag{34}$$

where $\beta = e^{-i\theta}$ is a unimodular complex number that depends on θ. Let us use the notation $[n]$ to denote the set $\{1,2,\ldots,n\}$ and let $|z|$ be the Hamming weight of z.

We could equivalently write $|G\rangle$ as

$$|G\rangle = \sum_{y\in 0}^{n} c_y \sum_{z\in H_y} |\tilde{z}\rangle\,, \tag{35}$$

where

$$|\tilde{z}\rangle = \frac{1}{\sqrt{2^n}} |z\rangle\,, \tag{36}$$

H_y is the set of all strings with Hamming weight y and

$$c_y = \beta^{\left(\sum_{|z|=y;i,j\in[n];i<j} z_i z_j\right)} = \beta^{y(y-1)/2}, \tag{37}$$

which, by symmetry, is the same for every $z \in H_y$. This is what we mean by Hamming weight symmetry.

Now, let us formally state the theorem and prove it.

Theorem 42. *For $k = n-1$, $p_x(G,\mathsf{U})$ can be computed in polynomial time for any choice of U and any $x \in \{0,1\}^n$.*

Proof. We first reduce computing $p_x(\mathsf{U})$, for a string $x \in \{0,1\}^n$ to computing $p_{0^n}(\mathsf{V})$, for a particular choice of V. To see this reduction, note that for a string $x \in \{0,1\}^n$, we can write $p_x(U_l)$ as

$$p_x(\mathsf{U}) = |\langle 0^n|\mathsf{V}|G\rangle|^2, \tag{38}$$

where

$$\mathsf{V} = U_1\mathsf{X}^{x_1} \otimes U_2\mathsf{X}^{x_2} \otimes \cdots \otimes U_n\mathsf{X}^{x_n}. \tag{39}$$

Then, observing that

$$p_x(\mathsf{U}) = p_{0^n}(\mathsf{V}), \tag{40}$$

we compute $p_x(\mathsf{U})$ using our ability to compute $p_{0^n}(\mathsf{V})$, for any V.

Now, consider computing the following quantity:

$$\mathsf{Z}[n] = \bigotimes_{i=1}^{n} \left(\langle 0|\cos\left(\frac{\theta_i}{2}\right)e^{-i\phi_i} + \langle 1|\sin\left(\frac{\theta_i}{2}\right)\right)|G\rangle\,. \tag{41}$$

Observe that

$$p_{0^n}(\mathsf{U}) = \left|\mathsf{Z}[n]\right|^2. \tag{42}$$

So, if we can compute $\mathsf{Z}[n]$, we can compute $p_{0^n}(\mathsf{U})$. In the next steps, we will show how to compute $\mathsf{Z}[n]$. Let

$$|\psi_{y,n}\rangle = c_y \sum_{z\in H_y} |\tilde{z}\rangle\,. \tag{43}$$

Note that

$$|G\rangle = \sum_{y\in 0}^{n} |\psi_{y,n}\rangle\,. \tag{44}$$

Additionally, define

$$\mathsf{Z}[n,y] = \bigotimes_{i=1}^{n-1} \left(\langle 0| \cos\left(\frac{\theta_i}{2}\right) e^{-i\phi_i} + \langle 1| \sin\left(\frac{\theta_i}{2}\right) \right) \left(\langle 0| \cos\left(\frac{\theta_n}{2}\right) e^{-i\phi_n} + \langle 1| \sin\left(\frac{\theta_n}{2}\right) \right) |\psi_{y,n}\rangle . \tag{45}$$

Note that

$$\begin{aligned} &\mathsf{Z}[n,y] \\ &= \cos\left(\frac{\theta_n}{2}\right) e^{-i\phi_n} \bigotimes_{i=1}^{n-1} \left(\langle 0| \cos\left(\frac{\theta_i}{2}\right) e^{-i\phi_i} + \langle 1| \sin\left(\frac{\theta_i}{2}\right) \right) |\psi_{y,n-1}\rangle \\ &+ \sin\left(\frac{\theta_n}{2}\right) \frac{c_y}{c_{y-1}} \bigotimes_{i=1}^{n-1} \left(\langle 0| \cos\left(\frac{\theta_i}{2}\right) e^{-i\phi_i} + \langle 1| \sin\left(\frac{\theta_i}{2}\right) \right) |\psi_{y-1,n-1}\rangle \\ &= \cos\left(\frac{\theta_n}{2}\right) e^{-i\phi_n} \mathsf{Z}[n-1,y] + \sin\left(\frac{\theta_n}{2}\right) \frac{c_y}{c_{y-1}} \mathsf{Z}[n-1,y-1]. \end{aligned} \tag{46}$$

Now, design a recursion tree, a part of which is shown in Fig. S1. We start with $\mathsf{Z}[n,y]$ and by multiplying each path with the corresponding path-weights, slowly recurse down the tree. The number of nodes never blows up because there is a lot of overlap during the recursion: for example, in Fig. S1, both $\mathsf{Z}[n-1,y]$ and $\mathsf{Z}[n-1,y-1]$ have the same daughter node $\mathsf{Z}[n-2,y-1]$, albeit with different path-weights.

The value of $\mathsf{Z}[n,y]$ is the total value we get in the following way.

- First, we multiply the path weights for each path.
- Then, we add up the paths.

The number of paths is

$$2+4+6+8+\cdots n = \mathcal{O}(n^2). \tag{47}$$

We set

$$\begin{aligned} &\mathsf{Z}[n,y] = 1, \;\; n < y, \\ &\mathsf{Z}[0,0] = 1, \\ &\mathsf{Z}[0,-1] = 1, \\ &\mathsf{Z}[1,0] = \cos\left(\frac{\theta_1}{2}\right) e^{-i\phi_1}. \end{aligned} \tag{48}$$

to deal with the base cases. Now, do this for every y and add the values to get $\mathsf{Z}[n]$. Since the tree has polynomially many paths that are polynomially large, it can be traversed in polynomial time by standard algorithms. □

C. Proof of Theorem 41 (Duality theorem)

Let G be an $n \times n$ grid graph. Consider the vertices a, b, and c as shown in Figure 3 of the main text (a is the red vertex, and b and c are the green vertices.)

Let $\overline{G}$ be the complement of the grid graph. Follow the following sequence of steps.

- Apply a local complementation on a (red vertex).
- Vertex delete a, b, and c.

What we get back is a resource state for MBQC because we can get an $(n-1) \times (n-1)$ grid graph from there, just by vertex deletion.

D. Proof of Theorem 39 (The hard regime)

For every $3 \leq k \leq n/2$, we construct a k–regular parent graph such that we can reach the hexagonal graph or the grid graph from that parent graph just by vertex deletion. If we can do that, then by Lemma 17 and Lemma 18, computing the probabilities of the parent graph would be GapP-hard, under inverse polynomial multiplicative precision. Then, the existence of a classical sampler indicates the collapse of the PH to $\mathsf{BPP}^{\mathsf{NP}}$, by Lemma 19.

1. *Technical constructions*

One idea would be to start with the grid-graph or the hexagonal lattice, and just "reverse engineer" a construction of the k–regular graph, by adding appropriate gadgets to every vertex. This indeed works for small values of k, as we see in a demonstration below.

Consider a family of n–vertex, k–regular graphs $\mathcal{F}$ and consider the following task.

$\mathcal{F}$-REGULARGRAPH$[n,k,x,\mathsf{U}]$

Input A description of an n–vertex, k–regular graph $G \in \mathcal{F}$, $x \in \{0,1\}^n$, and a description of the last layer of local rotations U.

Output An inverse polynomial multiplicative error estimate of $p_x(G,\mathsf{U})$.

Proposition 43. *There is an explicit $\mathcal{F}$ such that $\mathcal{F}$-*REGULARGRAPH$[n,3,x,\mathsf{U}]$ *is* GapP*-hard.*

Proof. Consider an n vertex hexagonal lattice. Then, we add periodic boundary conditions, which is equivalent

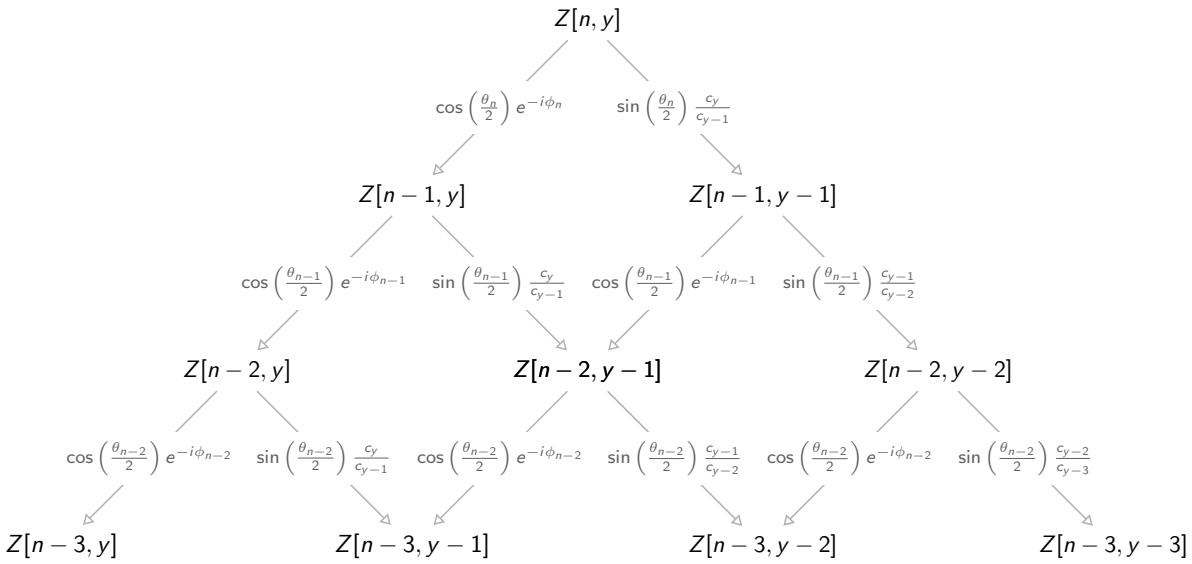

Figure S1. An illustration of how the recursion tree is constructed in the proof of Theorem 42, up to the first three levels.

to putting this on (or, in more colloquial terms, "wrapping it around") a torus. This makes it a 3–regular graph.

Note that one could recover a hexagonal lattice of $\Omega(\sqrt{n}) \times \Omega(\sqrt{n})$ vertices just by a sequence of vertex deletions to "cut open" the torus. The proof then follows from the observation that an n vertex hexagonal lattice is a resource state for MBQC. □

Proposition 44. *There is an explicit $\mathcal{F}$ such that $\mathcal{F}$-*REGULARGRAPH$\left[n, 4, x, \mathsf{U}\right]$ *is* GapP*-hard.*

Proof. The proof is the same as that of Proposition 43, except we start with an n–vertex grid graph, instead of the hexagonal lattice. The proof is also illustrated in Figure 2(a) of the main text. □

Proposition 45. *There is an explicit $\mathcal{F}$ such that $\mathcal{F}$-*REGULARGRAPH$\left[n, k, x, \mathsf{U}\right]$ *is* GapP*-hard, for $k = n - 5$ and $k = n - 4$.*

Proof. $\mathcal{F}$ is the complement of the graph families constructed in Proposition 43 (for $k = n - 4$) and Proposition 44 (for $k = n - 5$). In other words, the hard graphs are the complements of the hexagonal lattice or the grid graph, under closed boundary conditions. Using vertex deletion, one could "cut open" the boundary to reach the complement of a hexagonal lattice or a grid graph respectively, of side length $\Omega(\sqrt{n}) \times \Omega(\sqrt{n})$, which are resource states in MBQC, as proven in Theorem 41. □

For other hardness results, we need more involved constructions, with gadgets. Particularly, as explained in the main text, we need to invoke the Gale-Ryser theorem. The technical details of that theorem are provided below.

Proposition 46. *There is an explicit $\mathcal{F}$ such that $\mathcal{F}$-*REGULARGRAPH$\left[2m^2, k, x, \mathsf{U}\right]$ *is* GapP*-hard, for any $4 < k \leq m^2$.*

Proof. Start from two $m \times m$ grid graphs. Then, consider closed boundary conditions which is equivalent to putting them on a torus. Thereafter, have edges between the two tori to make every vertex k regular. This is illustrated in Figure 2(b) of the main text.

To argue that a valid connection pattern exists, we make use of the Gale-Ryser theorem, the details of which are given below. This suffices for our proof, because one could always delete one torus to reach an object that is a resource state for measurement based quantum computing, by Proposition 44.

We need to show that there is a way to connect the vertices such that the final graph is k–regular. To boost the regularity of each vertex of the two tori, on either side, to k, there are m^2 vertices that we need to add $k - 4$ extra edges to.

Since all the edges are "across" the two different grid graphs, and no edge is added "within" any grid, we can think of our situation as trying to construct a bipartite graph, with m^2 vertices on either side, such that the degrees on either side, given by A and B, are

$$\mathsf{A} = \mathsf{B} = \Big(\underbrace{k-4, \ldots, k-4}_{m^2 \text{ times}}\Big). \tag{49}$$

Now, we check the conditions of the Gale-Ryser theorem to prove that such degree sequences indeed correspond to a valid bipartite graph. Note that the proof of the Gale-Ryser theorem is constructive.

Although we do not explicitly construct the graph

here, we note that it can be easily done by following the steps of [52].

First condition

The first condition of the Gale-Ryser theorem is the following:

$$\sum_{i=1}^{m^2} a_i = \sum_{i=1}^{m^2} b_i, \tag{50}$$

where a_i is the i^{th} element of sequence A and b_i is the i^{th} element of sequence B.

Note that for our case, $a_i = b_i$ for every i. Hence, this statement is trivially true, as can be seen from (49). Now, let us state the second condition.

Second condition

For any $1 \leq p \leq m^2$, we need

$$\sum_{i=1}^{p} a_i \leq \sum_{i=1}^{m^2} \min(b_i, p). \tag{51}$$

We break the analysis of this condition into three cases, depending on the value of p.

First case

Let us take

$$1 \leq p \leq k-4. \tag{52}$$

Then, the RHS becomes pm^2. The LHS is upper bounded by $p(k-4)$. Hence, the RHS is larger for any $k \leq m^2$ and the second condition is satisfied.

Second case

Let us take

$$k-3 \leq p. \tag{53}$$

Then, the RHS is

$$(k-4)m^2. \tag{54}$$

The LHS is

$$(k-4)p. \tag{55}$$

Hence, the RHS is always greater than or equal to the LHS. To conclude, we have identified a way to connect two tori, each with m^2–vertices, together such that the resultant graph is k-regular, for any $4 \leq k \leq m^2$. □

Remark. *Let $n = 2m^2$. Then, by Proposition 44, $\mathcal{F}$-*REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$ *is* GapP*-hard, for any $4 < k \leq n/2$.*

Corollary 47. REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$ *is* GapP*-hard, for any $4 < k \leq n/2$.*

Now, we can use the duality theorem to argue the hardness of n–vertex, k–regular graphs with $n/2+1 \leq k \leq n-4$. Every hard graph for these families is a complement of the hard graphs we just explicitly constructed.

Proposition 48. *There is an explicit, efficiently constructible $\mathcal{F}$ such that $\mathcal{F}$-*REGULARGRAPH$\left[2m^2,k,x,\mathsf{U}\right]$ *is* GapP*-hard, for any $m^2+1 \leq k < 2m^2-5$..*

Proof. From Proposition 46, we have already seen how to construct an $\mathcal{H}$ such that $\mathcal{H}$-REGULARGRAPH$\left[2m^2,t,x,\mathsf{U}\right]$ is GapP-hard for $4 < t \leq m^2$. For a particular choice of m, let $H \in \mathcal{H}$ be a $2m^2$–vertex, t–regular graph, of the type that was constructed in the proof of Proposition 46. So, H comprises of two copies of a torus connected to each other in a certain way.

Let G be the complement of H. G is k–regular, where $k = 2m^2 - t - 1$. Then, by deleting all vertices on one copy of the torus, we can reach the complement of a grid graph from H, under closed boundary conditions. Then, we can "cut open" the boundary by vertex deletion to reach the complement of a grid graph whose side lengths are $\Omega(\sqrt{n}) \times \Omega(\sqrt{n})$. By Theorem 41, this is a resource state for measurement based quantum computation, and hence the proof follows. □

Remark. *Let $n = 2m^2$. Then, by Proposition 48, $\mathcal{F}$-*REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$ *is* GapP*-hard, for any $n/2 \leq k < n-5$.*

Corollary 49. REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$ *is* GapP*-hard, for any $n/2 < k < n-5$.*

Proposition 50. REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$ *is* GapP*-hard, for any $n/2 < k \leq n-5$.*

Finally, the proof of Theorem 39 follows from Proposition 43, Proposition 46, Proposition 45, Lemma 18, and Corollary 49.

E. Bounds on the entanglement width

We prove some conditional and unconditional lower bounds on the clique width of the graphs constructed in the previous section.

1. *Unconditional bounds*

Corollary 51. *Let G be any n–vertex k–regular graph constructed in the proofs of any one of Proposition 43, Proposition 44, Proposition 46, or Proposition 48. Then,*

$$\mathsf{cw}(G) = \Omega(\sqrt{n}). \tag{56}$$

Consequently,

$$\mathsf{ew}(|G\rangle) = \Omega(\log n). \tag{57}$$

Proof. For each G, we can reach either a grid graph or a hexagonal lattice by vertex deletion, or reach a complement of either. From [55, 58, 60], the n–vertex grid graph and the n–vertex hexagonal lattice have a clique width $\Omega(\sqrt{n})$. Since, by Lemma 34,

$$\mathsf{cw}(\overline{G}) \geq \frac{1}{2}\mathsf{cw}(G), \tag{58}$$

the complement of the grid and the hexagonal lattice also have clique width $\Omega(\sqrt{n})$.

By Lemma 35, since vertex deletion does not increase the clique width, the explicit graphs constructed in Proposition 46 and Proposition 48 also have clique width $\Omega(\sqrt{n})$. Consequently, the bound on entanglement width follows from Lemma 33 and Lemma 36. □

2. Conditional bounds

Corollary 52. *Let $\mathcal{F}$ be any k–regular graph family such that $\mathcal{F}$-REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$, is* GapP*-hard, for any $x \in \{0,1\}^n$. Then, assuming* BPP $\neq$ P$^{\#\mathsf{P}}$*,*

$$\mathsf{ew}(|G\rangle) = \omega(\log n), \tag{59}$$

for some $G \in \mathcal{F}$.

Proof. Assume the contrary. Then there is an algorithm to solve $\mathcal{F}$-REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$ in classical polynomial time using [17]. But, it was assumed that this task is GapP-hard. This implies BPP $=$ P$^{\#\mathsf{P}}$, which is a contradiction. □

By using a stronger conjecture, we can get a better lower bound.

Corollary 53. *Let $\mathcal{F}$ be any k–regular graph family such that $\mathcal{F}$-REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$, is* GapP*-hard, for any $x \in \{0,1\}^n$. Then, assuming* ETH *[**?**],*

$$\mathsf{ew}(|G\rangle) = \Omega\left(n^{\delta}\right), \tag{60}$$

for some $G \in \mathcal{F}$, and for some $\delta < 1$.

Corollary 54. *Let $\mathcal{F}$ be any k–regular graph family such that $\mathcal{F}$-REGULARGRAPH$\left[n,k,x,\mathsf{U}\right]$, is* GapP*-hard, for any $x \in \{0,1\}^n$ and every $G \in \mathcal{F}$ is reducible to a $\Omega(\sqrt{n}) \times \Omega(\sqrt{n})$ grid graph by a sequence of vertex deletions and local complementations. Then, assuming* SETH *[**?**],*

$$\mathsf{ew}(|G\rangle) = \Omega\left(n^{1/2}\right), \tag{61}$$

for some $G \in \mathcal{F}$.

The proofs are similar to that of Corollary 52. The $n^{1/2}$ comes from the fact that a $\Omega(\sqrt{n}) \times \Omega(\sqrt{n})$ grid graph "encodes" a GapP-hard probability on $\Omega(\sqrt{n})$-many "logical" qubits and the fact that entanglement width does not increase under vertex deletions and local complementation.

S4. CONNECTIONS TO PREVIOUS WORK

In this section, we discuss relations between our setup and what was considered in other relevant works.

A. A discussion of other known classical simulation methods

Apart from entanglement based methods discussed in the main text, there are known techniques to simulate quantum circuits based on the amount of other resources present in them, like T gates [29], stabilizer rank [30], or Wigner negativity [31, 32]. We had discussed in the main text why these methods are not adequate to classically simulate our setup: here, we go into more technical details.

In [29], the runtime of the classical simulator, for both probability estimation and sampling, scales as $\mathcal{O}\left(2^{ct}\right)$, where t is the number of T gates and c is a constant. For every fixed graph state $|G\rangle$, from Figure 1(a) of the main text, the presence of a last layer of unitaries means that the number of T gates, in the worst case, is n. Hence, classical simulation is inefficient.

From [30], any quantum circuit having Clifford gates and t T gates can be implemented by a quantum circuit starting with t magic states, each given by

$$|T\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi/4}|1\rangle\right). \tag{62}$$

Both the stabilizer and the approximate stabilizer rank of this equivalent circuit are $\mathcal{O}(2^{ct})$, for some constant c. Due to local rotations, this factor could be exponentially growing in n, in the worst case, for any graph state $|G\rangle$ in Figure 1(a) of the main text, rendering simulation inefficient. So far, it is not known how to prove better bounds on the stabilizer rank of a circuit or better utilize it as a resource in classical simulations.

Techniques based on Wigner negativity run into similar problems. Note that, when the circuit is only composed of Clifford gates, Wigner negativity, as defined in [31], is zero. Hence, techniques from Ref. [31] can be used to classically sample from the circuit. However, local rotations can significantly increase the negativity of the circuit, and the best known techniques for classically sampling from the circuit depend on a metric called "forward negativity" [32] being low. However, the authors in [32] demonstrated that the forward negativity can, in general, increase exponentially with the number of magic states, which rules out Wigner-negativity-based samplers for our setting.

For qubit circuits, an additional barrier is that qubit Wigner functions are much harder to define than their qutrit counterparts [33–35].

In conclusion, local rotations render a number of known classical simulation techniques useless for the simulation of graph states. Since entanglement is not

changed by local rotations, only entanglement-based simulation techniques seem to survive. This helps in our analysis and helps us to isolate entanglement as a potential cause of hardness.

B. A discussion of other entanglement measures

There are many measures to calculate both bipartite and multipartite entanglement, other than entanglement width, but we show below that it is harder to see what relation they have with hardness, if there is any relation at all. We discuss two popular measures here: the bipartitie von Neumann entropy and the geometric measure.

The bipartite von Neumann entropy, for a graph state $|G\rangle$, with respect to any bipartition (A,B) depends on the size, shape, and location of the bipartition and has the following upper and lower bounds [21]:

$$\log_2 r = S(|G\rangle_{\mathsf{A},\mathsf{B}}) \leq \mathsf{PauliPersistency}(|G\rangle) \leq \mathsf{minvertexcover}(G), \quad (63)$$

where Pauli Persistency is the number of local Pauli measurements required to fully disentangle the graph state, $\mathsf{minvertexcover}(G)$ is the minimal vertex cover of the corresponding graph G, and r is the Schmidt rank of $|G\rangle$ across the bipartition (A,B).

By a result from graph theory, for n–vertex, k–regular graphs,

$$\mathsf{minvertexcover}(G) \leq \frac{n\cdot k}{k+1}. \quad (64)$$

This implies

$$S\left(|G\rangle_{\mathsf{A},\mathsf{B}}\right) \leq \frac{n\cdot k}{k+1}. \quad (65)$$

Note that the upper bound grows with k, for a fixed n. However, for $k = n-1$, the von Neumann entropy is 1 across *any* bipartition [7], which indicates that the upper bound is not tight at all. To derive this property of the complete graph, we note that the complete graph reduces to a GHZ state under local Clifford rotations, and a GHZ state has von Neumann entropy equal to 1 for any bipartition.

On the other hand, for a worst case 2–regular graph, there is a bipartition (A,B) which achieves von Neumann entropy $n/2$. For example, this is achieved by n vertices arranged in a circle, where we fill A and B with alternate vertices. At the same time, a bipartition that divides the vertices into two semi-circles has constant von Neumann entropy. Therefore, this metric is very sensitive to the chosen bipartition. Since both the complete graph and any 2–regular graph are easy to classically simulate, but the former has low von Neumann entropy and the latter has high von Neumann entropy across the worst-case cut, it is not clear how von Neumann entropy of the worst case cut relates to classical simulation complexity.

Alternatively, one could consider von Neumann entropy across the best-case cut (A,B) subject to the constraint that $|\mathsf{A}| = \lfloor n/2\rfloor$. However, for every n, one can divide the n qubits into two decoupled sets A and B with $|\mathsf{A}| = \lfloor n/2\rfloor$ and define independent k-regular graphs on A and B, for any constant k. The proposed best-case von Neumann entropy would then be zero for all such graphs and will thus clearly not track the simulation complexity.

Therefore, if one wants to use the von Neumann entropy to track simulation complexity, one needs to make the choice of the cut more cleverly. Indeed, entanglement width, according to Eq. (21), is precisely a case of such a cleverly chosen cut: entanglement width is the minimum von Neumann entropy across the most tree-like bipartition, and we show how it tracks simulation complexity. It is plausible that one can find other such variants.

There are other measures, like the geometric measure of entanglement, given by

$$S_{\text{geom}}(|\psi\rangle) = -\log_2 \sup_{\alpha\in\mathcal{P}} |\langle\alpha|\psi\rangle|^2, \quad (66)$$

where $\mathcal{P}$ is the set of all separable states. For a graph G,

$$\mathsf{m_p} \leq S_{\text{geom}}(|G\rangle) \leq \mathsf{Pauli\ Persistency}(|G\rangle) \leq \mathsf{minvertexcover}(G) \leq \frac{n\cdot k}{k+1}, \quad (67)$$

where

$$\mathsf{m_p} = \max_{(\mathsf{A},\mathsf{B})} \mathsf{m_p}(\mathsf{A},\mathsf{B}). \quad (68)$$

In equation (68), (A,B) is a bipartition and $\mathsf{m_p}(\mathsf{A},\mathsf{B})$ is the maximum number of Bell pairs that can be created between (A,B), by a bipartite LOCC circuit, comprising only of CZ gates and local Clifford gates [26], when starting from $|G\rangle$. In other words, there can be CZ gates within each partition, but there should be no CZ gates across the bipartition. For a complete graph, $\mathsf{m_p}$ is 1.

Hence, the known upper and lower bounds on the geometric measure are also not tight, and it is not clear what relation, if any at all, these metrics have with hardness.

S5. COMMENTS ON AVERAGE CASE HARDNESS

In the summary section of the main text, we had pointed out that our results generalize to the average case. In this section, we discuss that in detail.

There could be two notions of average case for our setup:

- For a particular G, and random choice of U.

- For a random choice of G, and random choice of U, for a particular k–regular family.

For every hard k–regular graph identified in Theorem 39, for $3 \leq k \leq n-4$, computing $p_x(G, \mathsf{U})$ is GapP-hard with high probability over the choice of U, up to an additive error of $2^{-\mathcal{O}(m \log m)}$, using the worst-to-average case reductions in Ref. [73], where m is the number of gates of the circuit.

The results are not immediately extendable to a random choice of G, because the polynomial interpolation methods of [38, 39, 73] may take us beyond k–regular graph states. The question then becomes whether it is reasonable to expect average case hardness in this regime.

We assert such an expectation is reasonable and does not violate known results from graph theory, because *most* k–regular graphs, for a range of values for k, have $\Omega(n)$ clique width and $\Omega(\log n)$ entanglement width. This fact follows straightforwardly from techniques in graph theory, like Lemma 32, and from works like [58]. But we still state it formally and sketch a proof, just for the sake of completeness.

Note that this does not rule out classical samplers, as there could be a matching upper bound of $\mathcal{O}(\log n)$ for the entanglement width which would make efficient sampling possible by Lemma 22—but it makes this setting "almost" out of reach of known non-trivial, tree width based samplers.

Lemma 55. *Let G be an n–vertex, k–regular graph picked uniformly at random from the set of all possible n–vertex, k–regular graphs. Then*

$$\lim_{n\to\infty} \mathsf{Pr}[\mathsf{cw}(G) = \Omega(n)] = 1, \tag{69}$$

when $k = o(n)$, or $k = n - o(n)$. Consequently, under the same conditions as (69),

$$\lim_{n\to\infty} \mathsf{Pr}[\mathsf{ew}(|G\rangle) = \Omega(\log n)] = 1. \tag{70}$$

Proof. First, we prove that random k–regular graphs, for $k = o(n)$, have clique width $\Omega(n)$ with high probability. By Lemma 34, this means random $n-k-1$–regular graphs also have a clique width of $\Omega(n)$ with high probability, as these are complements of random k–regular graphs. The result on entanglement width then follows from Lemma 33 and Lemma 36, just as we have seen before.

Let $k = o(n)$. Let G be a random k–regular graph. It holds that

$$\mathsf{tw}(G) = \Theta(n), \tag{71}$$

with high probability [61]. From Lemma 32,

$$\mathsf{tw}(G) \leq 3 \cdot \mathsf{cw}(G) \cdot (t-1) - 1, \tag{72}$$

if G does not have the complete bipartite graph $K_{t\times t}$ as a subgraph. The proof then follows from the following result, which can be seen in Ref. [69]:

$$\lim_{n\to\infty} \mathsf{Pr}[K_{t\times t} \text{ is not a subgraph of } G] = 1, \tag{73}$$

for any constant t, for a random k–regular graph with $k = o(n)$. □

Note that Lemma 55 breaks down for the very specific case of when $k = \Theta(n)$. This is an artefact of the proof technique, because equation (73) breaks down for this case, and we did not find other simple ways to bound the clique width. Nonetheless, we conjecture that there should be a better way to bound the clique width for these cases.