

Entanglement bounds on the performance of quantum computing architectures

Zachary Eldredge^{1,2}, Leo Zhou³, Aniruddha Bapat^{1,2}, James R. Garrison^{1,2}, Abhinav Deshpande^{1,2},
Frederic T. Chong⁴ and Alexey V. Gorshkov^{1,2}

¹Joint Center for Quantum Information and Computer Science, NIST/University of Maryland, College Park, Maryland 20742, USA

²Joint Quantum Institute, NIST/University of Maryland, College Park, Maryland 20742, USA

³Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA

⁴Department of Computer Science, University of Chicago, Chicago, Illinois 60637, USA



(Received 24 October 2019; accepted 4 August 2020; published 26 August 2020)

There are many possible architectures of qubit connectivity that designers of future quantum computers will need to choose between. However, the process of evaluating a particular connectivity graph's performance as a quantum architecture can be difficult. In this paper, we show that a quantity known as the isoperimetric number establishes a lower bound on the time required to create highly entangled states. This metric we propose counts resources based on the use of two-qubit unitary operations, while allowing for arbitrarily fast measurements and classical feedback. We use this metric to evaluate the hierarchical architecture proposed by A. Bapat *et al.* [*Phys. Rev. A* **98**, 062328 (2018)] and find it to be a promising alternative to the conventional grid architecture. We also show that the lower bound that this metric places on the creation time of highly entangled states can be saturated with a constructive protocol, up to a factor logarithmic in the number of qubits.

DOI: [10.1103/PhysRevResearch.2.033316](https://doi.org/10.1103/PhysRevResearch.2.033316)

I. INTRODUCTION

As the development of quantum computers progresses from the construction of qubits to the construction of intermediate-scale devices, quantum information scientists have increasingly begun to explore various architectures for scalable quantum computing [1–4]. Researchers have quantified the cost imposed by moving from one architecture to another [5,6] and optimized the placement of qubits on a fixed architecture [7–9]. Experimentalists have also begun to test different architectures in laboratory settings [10,11].

In this work, we are interested in developing tools to evaluate the relative performance of different architectures. Here, “architecture” refers to the connectivity graph that defines the allowable elementary operations between qubits. We propose a natural metric based on entanglement measures. When several physical models are represented by a graph $G = (V, E)$, with a set of vertices V corresponding to qubits, and a set of weighted edges E corresponding to two-qubit operations (where the weights denote the maximum rates of operations), a useful metric is given by what we dub the “rainbow time,”

$$\tau_{\text{RB}}(G) = \max_{F \subset V, |F| \leq \frac{1}{2}|V|} \frac{|F|}{|\partial F|}, \quad (1)$$

where $|\partial F|$ denotes size of the boundary of F , i.e. the total weight of edges connecting F and $\bar{F} = V - F$.

We show that the rainbow time is a lower bound on the time required to create a highly entangled state on the graph (i.e., states of N qubits with $\mathcal{O}(N)$ bipartite entanglement). It is also the reciprocal of a well-studied graph quantity known as the isoperimetric number [12]. We note that this lower bound holds even when measurement and feedback are allowed to speed-up entanglement generation, such as in the case of Greenberger-Horne-Zeilinger states [13]. In contrast to Ref. [14], where architectures are evaluated assuming that only unitary operations are permitted, our results apply to the more general setting that allows nonunitary operations.

As a complementary result, we show that this lower bound is nearly tight—a procedure that distributes Bell pairs using maximum-flow algorithms nearly saturates this bound to produce $\mathcal{O}(N)$ entanglement across any bipartition, up to $\mathcal{O}(\log N)$ overhead. This suggests that beyond providing a bound, the rainbow time would be a useful witness to the speed at which entanglement can actually be generated.

II. PHYSICAL MODEL

In this paper, we evaluate the performance of quantum architectures with a connectivity graph given by G . Each vertex in the graph represents a single data qubit, and an edge exists between two vertices if two-qubit operations can be performed between them. We interpret the edge weight w_{ij} between vertices i and j as representing bandwidth, so that higher-weighted edges are capable of performing more two-qubit operations in a single unit of time.

We consider an example physical model where the edge weights represent the rate of distribution of entangled pairs

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

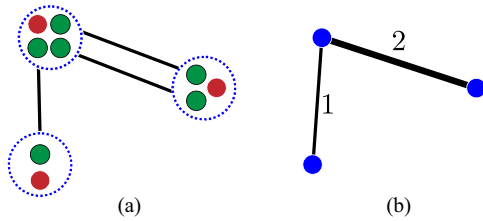


FIG. 1. Illustration of how a model with ancilla mediator qubits can be abstracted into one in which only data qubits and edge weights are tracked. In panel (a), each module (blue dashed circle) contains one data qubit (red) and several ancilla mediator qubits (green) that form Bell pairs with other modules. In panel (b), the module as a whole is represented by blue circles, while the ancilla mediator qubits are now represented by edge weights. Only the states of the data qubits are tracked.

as in Ref. [15]. Each vertex is a small module that contains a data qubit and some ancilla qubits. In each unit of time, Bell pairs are generated between the ancilla qubits on the edges of the graph, which can then be used to perform two-qubit gates on the data qubits [16,17]. The process of moving from this model to an abstracted connectivity graph is illustrated in Fig. 1. We assume that measurements, classical communication, and intra-module unitaries are arbitrarily fast, such that the bottleneck is given by quantum operations between modules. For example, this model can describe a trapped-ion system which uses photonic interconnects to generate entanglement between modules as in Refs. [18,19]. In this framework, vertex degrees and total graph edge weights represent required ancilla overheads, justifying their use as cost functions in Ref. [14].

While, for simplicity, we will focus in the main text on the above model, our results also apply to other physical models, up to constant-factor overheads. For example, since any two-qubit operation between data qubits can be performed by consuming two Bell pairs [20], the above model is equivalent to a model where edge weights are proportional to rates of two-qubit operations. In Appendix A, we show in more detail how to extend our results to this model, as well as to a model where edge weights represent coupling strengths in a Hamiltonian.

III. ENTANGLEMENT CAPACITY

Given a graph G , we wish to bound the total possible increase in a given entanglement measure after n rounds of entanglement distribution over its links. Suppose we fix a bipartition of the graph into two subgraphs supported on vertex subsets F and \bar{F} . We consider a general entanglement measure, S , which quantifies the bipartite entanglement between F and \bar{F} . We assume the following axioms: S is zero for product states $\rho_F \otimes \rho_{\bar{F}}$, additive between nonentangled regions, $S(\rho_{F\bar{F}} \otimes \tau_{F\bar{F}}) = S(\rho_{F\bar{F}}) + S(\tau_{F\bar{F}})$, and nonincreasing under local operations and classical communication. Entanglement measures that obey these axioms include the entanglement cost, the distillable entanglement, and the entanglement of formation [21,22]. All of these measures are identical to the von Neumann entropy for pure states.

By the result of Ref. [21], the entanglement after n rounds is bounded by n times the maximum single-round entanglement. We will therefore bound the entanglement generated in one round, going from ρ to ρ' . To produce ρ' , we begin with ρ and then generate entanglement on the graph edges. This means that w_{ij} ancilla Bell pairs are generated for each edge (i, j) crossing the boundary ∂F . The total number of Bell pairs is therefore $|\partial F|$, the sum over all the weights,

$$|\partial F| = \sum_{i \in F, j \in \bar{F}} w_{ij}. \tag{2}$$

Ignoring ancillas purely local to F or \bar{F} , the resulting state is $\rho \otimes \rho_{\text{Bell}}^{\otimes |\partial F|}$. The final state ρ' is then generated by local operations, assisted by classical communication, on this state. We denote the state that results from an arbitrary round of local operations and classical communications on ρ as $\text{LOCC}(\rho)$. Therefore, our axioms for S allow us to write

$$\begin{aligned} S(\rho') &= S[\text{LOCC}(\rho \otimes \rho_{\text{Bell}}^{\otimes |\partial F|})] \\ &\leq S(\rho \otimes \rho_{\text{Bell}}^{\otimes |\partial F|}) \\ &= S(\rho) + |\partial F| S(\rho_{\text{Bell}}), \\ \Rightarrow S(\rho') - S(\rho) &\leq |\partial F| S(\rho_{\text{Bell}}). \end{aligned} \tag{3}$$

Working in the units of $S(\rho_{\text{Bell}}) = 1$, we refer to this upper bound on the change in entanglement, $\Delta S \leq |\partial F|$, as the *entanglement capacity* of the (F, \bar{F}) bipartition in the graph G .

IV. RAINBOW STATES

We now define a highly entangled state whose creation serves as a benchmark for the performance of a quantum computing architecture.

Entanglement makes a useful benchmark for any quantum computer because it can be shown that computations that do not produce entanglement can be efficiently simulated classically [23–25]. Further motivation for producing highly entangled states can be found in quantum simulation, where a quantum simulator of general applicability ought to be capable of representing and simulating highly entangled states [26].

To select a particular entangled state for benchmarking, we consider “rainbow states.” In 1D contexts, for even N , a rainbow state is one in which qubits i and $N - i$ are maximally entangled [27,28]. The state itself is maximally entangled across a bipartition between the first $N/2$ qubits and the rest.

We extend this construction to arbitrary graphs. Suppose we consider a set of qubits V and any subset $F \subset V$, with the requirement that $|F| \leq \frac{1}{2}|V|$. Denote by F_i the i th vertex of F using an arbitrary ordering, and similarly use \bar{F}_i to index vertices in the complement \bar{F} . We can then define a “rainbow” state as one in which qubit F_i and qubit \bar{F}_i form a Bell pair, and any additional qubits in \bar{F} are left in the state $|0\rangle$. This state is illustrated for a particular choice of F and ordering in Fig. 2. Note that this construction is only well-defined if $|F| \leq \frac{1}{2}|V|$, as otherwise there will not be enough data qubits in \bar{F} to form Bell pairs with all the data qubits in F . The arbitrary ordering allows multiple rainbow states to be defined from the same F .

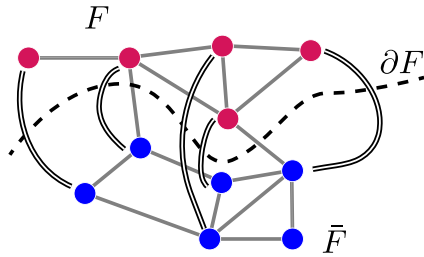


FIG. 2. An illustration of how a rainbow state is defined on an arbitrary subgraph F . Here, gray lines represent the connectivity graph of allowed two-qubit interactions, while doubled black lines represent maximally entangled qubit pairs. Qubits without a doubled line are assumed to be in state $|0\rangle$.

V. RAINBOW TIMES AND ISOPERIMETRIC NUMBER

Using the model for quantum architectures in which each edge weight of a graph G denotes the rate of entanglement generation across that edge, we can calculate the lower bound on the time required to create a rainbow state, according to the entanglement capacity. For any vertex subset F we define this time as

$$t(F) = \frac{|F|}{|\partial F|} = \frac{\text{number of qubits in } F}{\text{entanglement capacity of } (F, \bar{F})}. \quad (4)$$

As we have shown, the entanglement capacity corresponds to the total weight of edges across the boundary, which constrains the amount of entanglement that can be distributed to the subsystem F from its complement \bar{F} in unit time.

Although there are many choices for a highly entangled physical state associated with the subset F that would be hard to create, here we argue why the above metric $t(F)$ suffices for most considerations. Although there are many different states with $\mathcal{O}(N)$ entanglement which could be used to evaluate graphs, the rainbow state is easy to conceptualize and create. Since any bipartite entangled state can be converted either to or from Bell pairs through entanglement concentration or dilution [29], the rainbow state offers insight into the time required to create a general bipartite entangled state. Furthermore, rainbow states arise as ground states of novel models in condensed-matter physics [30], and thus the ability to create them can be important for quantum simulation. The difficulty to create rainbow states is also recognized in Ref. [13]. While there is freedom in defining a physical rainbow state via the pairing of vertices in F with those in \bar{F} , the precise choice of pairing does not affect the minimum time required to create the state according to the entanglement capacity, $t(F)$. While different rainbow states that share a common subset F may differ in how quickly they can be created, $t(F)$ serves as the common lower bound on the creation time for all of them, and thus we will focus on that metric here.

We will now use $t(F)$ to evaluate the quantum architecture G , the larger graph that contains F as a vertex subset. To do this, we find the maximum $t(F)$ given G . Note that this is not the same as maximizing entanglement entropy, which would simply yield half the graph without any consideration of the graph structure. Instead we ask: Of all the maximally entangled states we can build by bipartitioning V into F and \bar{F} , which of them is slowest to build according to the

entanglement capacity? We call the associated quantity $t(F)$ the *rainbow time* of the graph G and denote it $\tau_{\text{RB}}(G)$, as defined in Eq. (1).

The rainbow time has a simple and attractive interpretation, can be directly connected to quantum computing tasks, and is applicable to various physical models of computation. In addition, it can be directly connected to a quantity known as the isoperimetric number $h(G)$ [12], sometimes also known as the Cheeger constant, which is well-studied in graph theory and computer science [31–33]. As we have defined it, the rainbow time is simply $\tau_{\text{RB}}(G) = 1/h(G)$ [34]. Thus, aiming to minimize the rainbow time (so that large entangled states can be easily created) in a quantum architecture is equivalent to maximizing the isoperimetric number. An “isoperimetric set” is a vertex subset F that achieves $t(F) = \tau_{\text{RB}}(G)$. Often, isoperimetric numbers appear in the context of expander graphs, which are constructed to possess large isoperimetric numbers [35] and are used to prove important results in complexity theory [36–38]. Intuitively, a small isoperimetric number (large τ_{RB}) means that a graph has bottlenecks, and a sizable subset can easily be disconnected by removing relatively few edges. This also implies that an architecture with large τ_{RB} is more prone to becoming disconnected due to the failure of a small number of edges.

Even though computation of the exact rainbow time is NP-hard for general graphs [12], it can be approximated to within an $\mathcal{O}(\sqrt{\log N})$ factor [39]. There are also efficiently computable bounds on the rainbow time, including ones using the eigenvalues of the graph Laplacian [12]. Furthermore, for many specific graphs, we can evaluate the rainbow time efficiently. In Appendix B, we have done this for the complete, star, and grid graphs, as well as the hierarchical products and hierarchies presented in Ref. [14]. In particular, we compare hierarchies to d -dimensional grids and show that, for some parameters, hierarchies have lower rainbow time and lower total edge weight than grids, making them promising architectures for quantum computing.

VI. CREATING RAINBOW STATES

So far we have shown that rainbow time τ_{RB} serves as a lower bound for generating maximum entanglement across any bipartition of the system. We now examine whether this bound can be saturated, in the sense that one can create a rainbow state across any bipartition in time $\tilde{\mathcal{O}}(\tau_{\text{RB}})$. We will show that for a general graph, there is an explicit protocol that prepares a rainbow state in time no more than $\lceil \tau_{\text{RB}} \ln |F| \rceil$ for any bipartition where F is the smaller subset, indicating that the bound τ_{RB} is tight up to a logarithmic factor.

We begin the proof by mapping the problem of creating rainbow state to the MaxFlow problem in computer science [40]. Here, we restrict our attention to quantum architectures on graph $G = (V, E)$, where the edge weights are integers that represent the number of Bell pairs that can be generated across the edge per unit time. Suppose we are given arbitrary vertex subsets F and K , where $|F| = |K| \leq |V|/2$, and $K \subset \bar{F}$. To create a Bell state between a given pair of nodes in a single time step, we can specify a path connecting them on the graph G , generate Bell pairs on each edge along that path, and then perform entanglement connection on each internal node to

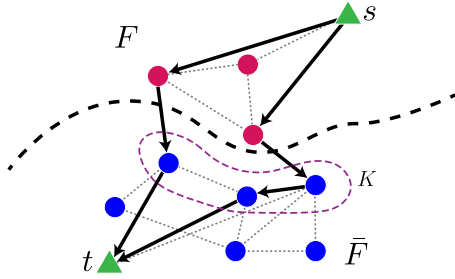


FIG. 3. An illustration of the fictitious nodes added to the isoperimetric set, F , and a set of equal size K (encircled by purple dashed line), to create a flow network. The new fictitious nodes, s and t , appear as green triangles connected to every node in F and K , respectively; the original nodes and edges are pink (in F) and blue (in \bar{F}) circles. The edges have weight one. The flow, shown by arrows, transfers $\lceil |F|/\tau_{\text{RB}} \rceil = 2$ units of entanglement across the bipartition. Gray, dotted edges are not used by the flow.

convert the string of Bell pairs into one long-distance Bell pair. We can create many distant Bell pairs in this way during a single time step by specifying many paths. However, the set of paths must not use any edge more often than the weight of that edge allows for, since by definition the weight of an edge limits the number of Bell pairs the edge can generate in a unit time step. Thus, we can interpret the weight of each edge as its *capacity*, and the collection of paths as a *flow* of entanglement from F to K , as illustrated in Fig. 3. Suppose we now attach a fictitious source node s to each node in F , and a fictitious sink node t to every node in K . Then the problem of maximizing the number of Bell pairs simultaneously generated between F and K is the same as the problem of maximizing the flow from the source s to the sink t . The latter problem is known as MaxFlow, visualized in Fig. 3, and an explicit protocol to give the maximum possible amount of flow can be found efficiently via, e.g., the Ford-Fulkerson algorithm [41]. Note that if all the edge weights are integers, a flow of maximum value exists in which the flow carried by each edge is also an integer [42].

To demonstrate that a flow approach yields an efficient creation of a rainbow state, we invoke the MaxFlow-MinCut theorem, which says that the maximum flow has the same value as the minimum cut [40]. Here, a “cut” means a bipartition of the graph separating s and t , and its value is the total weight of all edges that cross the bipartition. By finding a lower bound on the value of all possible cuts in a graph, we show that a flow larger than or equal to this bound must exist.

Suppose that we now consider any cut of the graph into some arbitrary pair of subsets $\{s\} \cup S$ and $\{t\} \cup T$. The boundary of this cut will consist of edges from $s \rightarrow T$, $S \rightarrow t$, and $S \rightarrow T$. Its magnitude can be written as

$$|\text{Cut}(S, T)| = |T \cap F| + |S \cap K| + |\partial S|, \quad (5)$$

since s and t are connected only to nodes in F and K , respectively, and the edges in $S \rightarrow T$ are just the boundary of S in the original graph. To evaluate $|\partial S| = |\partial T|$, we will assume that $|S| \leq \frac{1}{2}|V|$, meaning we can apply the isoperimetric condition $|S| \leq |\partial S|\tau_{\text{RB}}$. (If this is not the case, then a near-identical argument can be made applying this condition to T .) To account for cases where $\tau_{\text{RB}} < 1$, we will write this

as $|\partial S| \geq m|S|$ where $m = \min(1, 1/\tau_{\text{RB}})$. We then note that

$$\begin{aligned} |\partial S| &\geq m|S| \geq m(|S \cap F| + |S \cap K|) \\ &\geq m(|F| - |T \cap F| + |S \cap K|). \end{aligned} \quad (6)$$

By inserting this lower bound for $|\partial S|$ into Eq. (5), we obtain

$$|\text{Cut}(S, T)| \geq (1 - m)|T \cap F| + (1 + m)|S \cap K| + m|F|. \quad (7)$$

Since we know $m \leq 1$, we obtain the final bound on the cut magnitude,

$$|\text{Cut}(S, T)| \geq m|F|. \quad (8)$$

If $m = 1$ (i.e., $\tau_{\text{RB}} \leq 1$), then it follows that the value of the smallest cut is greater than $|F|$, meaning that a flow exists of magnitude at least $|F|$, which creates the rainbow state in a single round. If $m < 1$ (i.e., $\tau_{\text{RB}} > 1$), then we find that a flow exists of magnitude $|F|/\tau_{\text{RB}}$ [43]. Once $|F|/\tau_{\text{RB}}$ nodes are entangled, they can be disconnected from s and t , and the process repeated on a new set of nodes $F_1 \subset F$. Therefore, after n rounds of computation, the remaining set of nodes waiting for entanglement F_n is produced by removing $1/\tau_{\text{RB}}$ of the nodes in set F_{n-1} , with $F_0 = F$, allowing us to compute the maximum size of F_n inductively:

$$\begin{aligned} |F_n| &\leq \left(1 - \frac{1}{\tau_{\text{RB}}}\right) |F_{n-1}| \\ &\leq \left(1 - \frac{1}{\tau_{\text{RB}}}\right)^n |F| < e^{-n/\tau_{\text{RB}}} |F|. \end{aligned} \quad (9)$$

Once $|F_n| < 1$, the process is complete, as there are no fractional nodes. It follows that $\lceil \tau_{\text{RB}} \ln |F| \rceil$ rounds suffice to complete the entangling process.

VII. OUTLOOK

In this work, we have presented a new metric for evaluating proposed architectures for quantum computers. While we have proven that any vertex subset F can have a rainbow state prepared in $\lceil \tau_{\text{RB}} \ln |F| \rceil$ time, test simulations on many example small graphs suggest that flow-based algorithms can create rainbow states in $\lceil \tau_{\text{RB}} \rceil$ time. It is thus possible that the logarithmic factor can be removed and that the rainbow time lower bound is fully tight and saturable. In addition, although our argument suggests that for any bipartition of the system, *there exists a rainbow state* that can be created in $\lceil \tau_{\text{RB}} \ln |F| \rceil$ time, other rainbow states (where the connections between node pairs are permuted) may take longer. It would be interesting to upper bound the creation time of arbitrary rainbow states using tools from classical network theory such as routing time [44,45].

Finally, another open question is how the entanglement capacity, used here in terms of the rainbow time, can be applied to the analysis of quantum algorithms. While the rainbow time is not enough to provide an upper bound on the time-complexity of running a quantum algorithm on a given quantum architecture, it can provide a lower bound when the amount of entanglement required in the algorithm is known. References [46,47] explore the question of how entanglement grows during Shor’s algorithm and in adiabatic quantum computing. These complement other results showing

that low-entanglement systems can be simulated efficiently on a classical computer [23,48]. Rainbow time can also be used to benchmark algorithms for compilation and gate decomposition of quantum circuits, by comparing their realized circuit depth to this theoretical minimum required time.

ACKNOWLEDGMENTS

We thank A. Childs, A. Harrow, L. Jiang, D. Leung, G. Smith, and X. Wu for discussions. Z.E., A.B., J.R.G., A.D., and A.V.G. acknowledge funding by ARO MURI, DoE ASCR Quantum Testbed Pathfinder program (Award No. DE-SC0019040), AFOSR, ARL CDQI, NSF PFCQC program, AFOSR MURI, DoE BES Materials and Chemical Sciences Research for Quantum Information Science program (Award No. DE-SC0019449), DoE ASCR Accelerated Research in Quantum Computing program (Award No. DE-SC0020312), and NSF PFC at JQI. A.B. is supported in part by the QuICS Lenczos Fellowship. Z.E. is supported in part by the ARCS Foundation. L.Z. is supported in part by the National Science Foundation and the Center for Ultracold Atoms. J.R.G. was supported in part by the NIST NRC Research Postdoctoral Associateship Award. F.C. is funded in part by EPiQC, an NSF Expedition in Computing, under grant CCF-1730449; in part by STAQ, under grant NSF Phy-1818914; and in part by DOE Grants No. DE-SC0020289 and No. DE-SC0020331. This work was performed in part at the Aspen Center for Physics, which is supported by National Science Foundation Grant No. PHY-1607611.

APPENDIX A: ENTANGLEMENT CAPACITIES ON VARIOUS PHYSICAL MODELS

In this Appendix, we will derive the entanglement capacity for several different physical models that can correspond to a graph. Consider a graph, G , and select a subset of the vertices, F . We then want to show that the maximum amount of entanglement that can be created between F and \bar{F} in unit time is proportional to the size of the boundary, $|\partial F|$. We will allow arbitrary constant factors, and discuss how this bound arises in two different physical situations. As in the main text, we consider entanglement measures S on two regions so long as S obeys the following rules:

(1) Additively distributive over the tensor product, so $S(\rho \otimes \sigma) = S(\rho) + S(\sigma)$ if ρ and σ are supported on both sides of the bipartition.

(2) Zero for states which are a product of states on each region, $S(\rho_F \otimes \rho_{\bar{F}}) = 0$.

(3) Nonincreasing after any operation which is local to each region, even if we permit classical communication.

In the main text, we showed how to apply these axioms to the analysis of a case in which computation was performed by the production and consumption of Bell pairs. Here we also look at a gate model of computation and a case in which the graph describes the limits on a time-dependent interaction Hamiltonian.

1. Unitaries

In this model, each graph edge of weight w_{ij} represents the capability to perform w_{ij} unitaries between qubits i and

j in a time step. These unitaries are freely chosen by the experimenter. For two qubits, the ability to apply multiple unitaries is no different from the ability to apply an arbitrary unitary. However, we are considering cases where the qubits are part of a larger system, meaning we may wish to perform unitaries in sequence on different pairs to perform a more complicated computation.

We note that every two-qubit unitary can be performed using two Bell pairs as a shared resource and applying local operations. This can be easily seen in the following process:

(1) Alice and Bob start with a data qubit each and two Bell pairs shared between them. They wish to implement an arbitrary two-qubit unitary using only local operations and classical control.

(2) Alice uses one Bell pair and classical communication to teleport her qubit to Bob.

(3) Bob uses his local operations to perform the desired two-qubit gate.

(4) Bob teleports Alice’s qubit back to her.

Therefore, the state ρ' can be obtained from the state ρ by using local operations and classical communication (LOCC) and consuming up to $2|\partial F|$ Bell pairs in the process. Since LOCC cannot increase S , it follows that

$$S(\rho') \leq S(\rho \otimes \rho_{\text{Bell}}^{\otimes 2|\partial F|}) \tag{A1}$$

$$\Rightarrow \Delta S \leq 2|\partial F|S(\rho_{\text{Bell}}). \tag{A2}$$

This suggests that the ability to perform arbitrary unitaries is up to twice as powerful as the ability to distribute arbitrary Bell pairs, which makes sense, as an arbitrary two-qubit gate cannot necessarily be performed with one Bell pair (for instance, SWAP requires two) [20]. Two Bell pairs, however, suffice to implement any arbitrary two-qubit unitary. In any case, this still yields an entanglement capacity $\Delta S = \mathcal{O}(|\partial F|)$ bound as desired.

2. Hamiltonians

We will now consider a case in which the graph describes a Hamiltonian, possibly time-dependent. The graph will restrict the strength of these Hamiltonians. If we assume that $G = (V, E)$, then the Hamiltonian can be written as a sum over the two-qubit operations:

$$H(t) = \sum_{(i,j) \in E} h_{ij}(t). \tag{A3}$$

We then impose the condition

$$\forall t : \|h_{ij}(t)\| \leq w_{ij}, \tag{A4}$$

where w_{ij} is the i - j edge weight. We can then apply the “small incremental entangling” (SIE) theorem [49]. In particular, we apply the special case used in Ref. [50] to bound the total amount of entanglement generated by this Hamiltonian. If H is a sum of pairwise Hamiltonians h_{ij} acting on qubits, then the time-rate of entanglement generation on a set F of sites is

$$\left| \frac{dS_F}{dt} \right| \leq 36 \log(2) \sum_{i \in F, j \in \bar{F}} \|h_{ij}\|. \tag{A5}$$

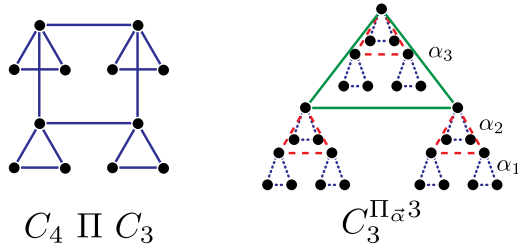


FIG. 4. Examples of a hierarchical product (left) and a weighted hierarchy (right).

Here, S_F is the von Neumann entropy of the reduced density matrix on the region F . This can be derived from Eq. (3) of Ref. [50], and specifying two-body terms and qubit sites, but the result could be extended to qudits or general k -body interactions. The sum over Hamiltonian norms, in the graph context, corresponds to a sum over graph edges. Since every Hamiltonian strength is limited by the corresponding edge weight, $\sum \|h_{ij}\| \leq \sum w_{ij} = |\partial F|$. Therefore, we can specifically say that for this case, $\Delta S_F = \mathcal{O}(|\partial F|)$. Many other entanglement measures, such as entanglement of formation or entanglement cost, can be related to the von Neumann entropy [22]. In particular, many entanglement measures on mixed states can be defined as a weighted sum over pure state components; since none of the pure states can increase dramatically in entanglement under this process, the entanglement measure on the mixed state is similarly limited.

APPENDIX B: APPLICATION TO HIERARCHICAL PRODUCT AND HIERARCHIES

In this Appendix, we calculate the rainbow times for the hierarchical products and hierarchies of Ref. [14]. A hierarchical product is a graph product denoted $G \Pi H$ in which $|G|$ copies of H are connected at their root (first) vertices by the graph G . By iterating this process, we can create a hierarchy, in which higher-level graphs connect lower-level identical subhierarchies. We also extend this concept to that of a weighted hierarchy, in which the edges on level i have weight α_i . We write a k -level hierarchy with a vector of weights $\vec{\alpha}$ as $G^{\Pi_{\vec{\alpha}} k}$, where G is the base graph. Finally, if $\alpha_i = \alpha^{i-1}$, so that edge weight scales geometrically with the level of the hierarchy, then we simply write $G^{\Pi_{\alpha} k}$. Some examples are shown in fig. 4.

To calculate the rainbow time for a hierarchical product, we make use of the result from Ref. [12] that there must exist an isoperimetric set [a vertex set F such that $\tau(F) = \tau_{RB}(F)$] that is connected and whose complement \bar{F} is connected. Therefore, we will look at all possible subgraphs of $H_1 \Pi H_2$ where both F and \bar{F} are connected. From these, we will search for the one with the largest $\tau(F)$. Since some isoperimetric set is guaranteed to exist in this set of subgraphs, this maximization over $\tau(F)$ in this set will also give us $\tau_{RB}(H_1 \Pi H_2)$. We will begin by specifying three cases, illustrated in fig. 5. These cases cover all possible subsets with the right connectedness properties and therefore allow us to find the maximizing set for the graph and $\tau_{RB}(H_1 \Pi H_2)$.

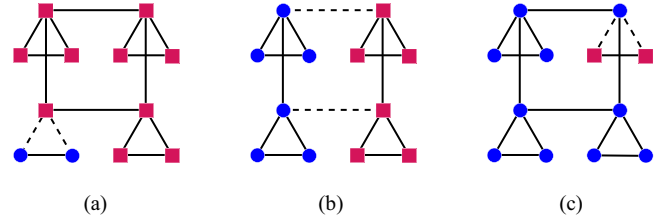


FIG. 5. Three classes of subgraph used in our proof. Circles represent vertices in F , squares are vertices in \bar{F} , and dashed lines are edges in ∂F . (a) A situation in which part of one copy of H_2 is in F . (b) A situation in which the division between F and \bar{F} lies entirely in H_1 . (c) A situation in which all but one of the copies of H_2 are entirely contained in F .

One such set would cover part of one copy of H_2 . However, note that if the root vertex of H_2 were included in F , then we would have to include all the descendants of H_2 , since otherwise \bar{F} would not be connected. Therefore, this class will only include subsets of H_2 which do not include the root vertex. In this case, we must maximize over all possible subsets of H_2 to find the maximum $\tau(F)$. This may seem like it would yield $\tau_{RB}(H_2)$; however, in this instance we can pick subsets of H_2 which make up a majority of H_2 , which is not allowed for τ_{RB} . We define the *unrestricted rainbow time* as

$$u_{RB}(G) = \sup_{F \subset G \setminus G_1} \tau(F). \tag{B1}$$

Here, $G \setminus G_1$ refers to G with its first vertex removed. Therefore, any set from this class will offer a candidate rainbow time of at most $\tau(F) = u_{RB}(H_2)$.

The second class of candidate sets would cross one or more copies of H_2 . Since F must be connected, the path between these copies must be included in F , which means the root vertices of each H_2 that connect to each other via H_1 must also be in F . Then, as shown above, the entire copy of H_2 must be included. As a result, this case is equivalent to choosing copies of H_2 and either entirely including them in F or entirely excluding them. This problem reduces to dividing up H_1 , and then calculating as if each vertex had an effective volume of $|H_1|$. Therefore, we can find the maximum $\tau(F)$ of these sets by simply finding $\tau_{RB}(H_1)$ and scaling it by $|H_2|$.

The final class of sets F which meets the connectedness criteria would be an F which includes all of H_1 and then all but one copy of H_2 completely, with perhaps some of the remaining H_2 also included. However, this F would necessarily be larger than half of the total graph $H_1 \Pi H_2$, and therefore we can discard it as a candidate set for determining the rainbow time. We combine the first two options and conclude that

$$\tau_{RB}(H_1 \Pi H_2) = \max [u_{RB}(H_2), |H_2| \tau_{RB}(H_1)]. \tag{B2}$$

We now seek to apply this to hierarchies $G^{\Pi_{\alpha} k}$. Just as before, if a vertex is included in F , then we must also include in F all its descendants in the hierarchy; otherwise, the complement \bar{F} will not be connected. Therefore, all bipartitions can be reduced to choosing a particular level of the hierarchy to cut—on that level, either a vertex will be included or not included, and this must apply to all of its descendants as well. Every bipartition can then be mapped to a bipartition of G , but one where every vertex is scaled by $|G|^{i-1}$ due to the size of

each subhierarchy [note that the large number of vertices not in F do not contribute to $\tau(F)$]. In addition, $\tau(F)$ must also be modified by the edge weight, which we define to be α_i on level i .

There is one important difference between the top (k th) level and all others, which arises from the constraint that $|F| \leq \frac{1}{2}|G^{\Pi_{\alpha^k}}|$. A cut on the top level must not include more than half of the highest-level copy of G , while all lower levels can use any cut at all as long as it does not include the root vertex. Whatever level we cut, the cut depends only on the base graph G , with each node standing for $|G|^{i-1}$ total nodes below it. Therefore, we can write the overall τ_{RB} as a maximization over these options:

$$\tau_{RB}(G^{\Pi_{\alpha^k}}) = \max \left(\frac{|G|^{k-1}}{\alpha_k} \tau_{RB}(G), \sup_{i < k} \frac{|G|^{i-1}}{\alpha_i} u_{RB}(G) \right). \tag{B3}$$

For specificity, we will evaluate the case where $G = K_n$, the complete graph, and $\alpha_i = \alpha^{i-1}$, which was proposed in Ref. [14] as an architecture. Here, the maximization over lower levels [the second term in Eq. (B3)] can be reduced to either to the first level or the $k - 1$ level, since we simply have to pick the largest element in a geometric sequence defined by n/α . We can write the resulting maximization as a choice between three options,

$$\tau_{RB}(K_n^{\Pi_{\alpha^k}}) = \max \left[1, \left(\frac{n}{\alpha}\right)^{k-1} \frac{2}{n}, \left(\frac{n}{\alpha}\right)^{k-2} \right]. \tag{B4}$$

Whereas one might have expected two options to arise (cut at the top or at the bottom), we actually have three. For $\alpha > n$, the edges grow in capacity too quickly for the increased volume to make a higher-level cut worthwhile, so the optimal cut is at the bottom, yielding a constant scaling with n . Two other options appear at $n > \alpha$, where cutting higher up the hierarchy allows for greater volume of qubits in F without too much penalty caused by changing edge weights. The reason there are two strategies is that it may be possible to cut a larger portion of a lower hierarchy and exploit the split between τ_{RB} and u_{RB} . [For K_n , in particular, the cut that includes all but the root vertex satisfies $u_{RB}(K_n)$.]

TABLE I. Important statistics for graphs. Here, only the asymptotic scaling with N is written. In addition to the rainbow time τ_{RB} for each graph, we also include the total weight of all edges w , and the maximum graph degree Δ . Rainbow times for graphs other than hierarchies can be found in terms of isoperimetric number in Refs. [12,32].

| Graph Name | τ_{RB} | w | Δ |
|------------------------|----------------------------------|------------------------------|------------|
| K_N | N^{-1} | N^2 | N |
| S_N | 1 | N | N |
| d -dimensional Grid | $N^{1/d}$ | N | $2d$ |
| $K_n^{\Pi_{\alpha^k}}$ | $N^{\max(0, 1 - \log_n \alpha)}$ | $N^{\max(1, \log_n \alpha)}$ | $\log_n N$ |

To place these results in context, we compare the rainbow time of $K_n^{\Pi_{\alpha^k}}$ to the total rainbow time of other graphs. To do this, we write the rainbow time in terms of the total number of qubits in a graph, N , and concern ourselves with the overall scaling. For the purpose of comparison, we consider hierarchies where the number of levels scale logarithmically as $k = \log_n N$, while α, n are constant parameters independent of N . In this language, $\tau_{RB}(K_n^{\Pi_{\alpha^k}}) = \Theta(N^{\max(0, 1 - \log_n \alpha)})$. We compare this to the rainbow time of some other graphs in Table I. References [12,32] give the isoperimetric number for K_N, S_N (the star graph of N nodes) and grids (which are Cartesian products of paths). Satisfying sets for these graphs are: for K_N and S_N , an arbitrary half of the nodes; for grids, a hypercube placed in one corner that takes up half the total volume.

One goal would be to identify a set of parameters where a hierarchy outperforms a d -dimensional grid architecture. We are most concerned with comparing to the d -dimensional grid because the other candidates we present, K_N and S_N , both have very large degree, making them impractical for scalable architectures, although both have been used for small quantum devices [10]. We find that the rainbow time of the hierarchy with base graph K_n and scaling constant α will be better (smaller) than that of the grid if $\alpha > n^{(d-1)/d}$. If it also holds that $n > \alpha$, then the hierarchy will accomplish this with a total edge weight scaling identically as the grid. It is possible to achieve a smaller prefactor in this scaling under a suitable choice of n, α ; for example, when $d = 2$, the choice of $n = 3, 4$ and $\alpha = n^{1-1/d}$ gives lower total edge weight for the hierarchy than the grid. We conclude that a hierarchy $K_n^{\Pi_{\alpha^k}}$ with $\alpha \in [n^{1-1/d}, n)$ has both lower rainbow time and lower total edge weight than a d -dimensional grid of qubits.

[1] C. Monroe and J. Kim, *Science* **339**, 1164 (2013).
 [2] M. Ahsan and J. Kim, in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE'15)* (IEEE Conference Publications, Washington, D.C., 2015), pp. 1108–1113.
 [3] A. Pirker, J. Walln ofer, and W. D ur, *New J. Phys.* **20**, 053054 (2018).
 [4] B. Villalonga, S. Boixo, B. Nelson, C. Henze, E. Rieffel, R. Biswas, and S. Mandr a, *npj Quantum Inf.* **5**, 86 (2019).

[5] D. Cheung, D. Maslov, and S. Severini, in *Proceedings of the Workshop on Quantum Information* (2007).
 [6] A. Holmes, S. Johri, G. Guerreschi, J. S. Clarke, and A. Y. Matsuura, *Quantum Sci. Technol.* **5**, 025009 (2020).
 [7] D. Rosenbaum and M. Perkowski, in *Proceedings of the 40th IEEE International Symposium on Multiple-Valued Logic* (IEEE, Washington, D.C., 2010), pp. 270–275.
 [8] D. J. Rosenbaum, in *Proceedings of the 8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC'13)*, Leibniz International Proceedings in

- Informatics (LIPIcs), Vol. 22, edited by S. Severini and F. Brandao (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013), pp. 294–307.
- [9] M. Pedram and A. Shafaei, *IEEE Circuits Syst. Mag.* **16**, 62 (2016).
- [10] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, *Proc. Natl. Acad. Sci. USA* **114**, 3305 (2017).
- [11] D. Maslov, *New J. Phys.* **19**, 023035 (2017).
- [12] B. Mohar, *J. Combin. Theory, Ser. B* **47**, 274 (1989).
- [13] C. Meignant, D. Markham, and F. Grosshans, *Phys. Rev. A* **100**, 052333 (2019).
- [14] A. Bapat, Z. Eldredge, J. R. Garrison, A. Deshpande, F. T. Chong, and A. V. Gorshkov, *Phys. Rev. A* **98**, 062328 (2018).
- [15] K. S. Chou, J. Z. Blumoff, C. S. Wang, P. C. Reinhold, C. J. Axline, Y. Y. Gao, L. Frunzio, M. H. Devoret, L. Jiang, and R. J. Schoelkopf, *Nature* **561**, 368 (2018).
- [16] D. Gottesman and I. L. Chuang, *Nature* **402**, 390 (1999).
- [17] L. Jiang, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, *Phys. Rev. A* **76**, 062323 (2007).
- [18] K. R. Brown, J. Kim, and C. Monroe, *npj Quantum Inf.* **2**, 16034 (2016).
- [19] R. Nigmatullin, C. J. Ballance, N. de Beaudrap, and S. C. Benjamin, *New J. Phys.* **18**, 103028 (2016).
- [20] J. Eisert, K. Jacobs, P. Papadopoulos, and M. B. Plenio, *Phys. Rev. A* **62**, 052317 (2000).
- [21] C. H. Bennett, A. W. Harrow, D. W. Leung, and J. A. Smolin, *IEEE Trans. Inf. Theory* **49**, 1895 (2003).
- [22] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, *Rev. Mod. Phys.* **81**, 865 (2009).
- [23] G. Vidal, *Phys. Rev. Lett.* **91**, 147902 (2003).
- [24] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, *Phys. Rev. Lett.* **93**, 207204 (2004).
- [25] Although universal quantum computation is possible in the limit of vanishing entanglement by implementing any quantum circuit \mathcal{C} in a way that's controlled by a qubit in the state $\sqrt{1-\epsilon}|0\rangle + \sqrt{\epsilon}|1\rangle$ [51], such computation still requires the ability to implement the circuit \mathcal{C} . This means that any entanglement-based bound on the time-complexity of implementing \mathcal{C} would still apply to the ϵ -entangled version.
- [26] J. I. Cirac and P. Zoller, *Nat. Phys.* **8**, 264 (2012).
- [27] G. Ramírez, J. Rodríguez-Laguna, and G. Sierra, *J. Stat. Mech.* (2015) P06002.
- [28] R. N. Alexander, A. Ahmadain, Z. Zhang, and I. Klich, *Phys. Rev. B* **100**, 214430 (2019).
- [29] C. H. Bennett, H. J. Bernstein, S. Popescu, and B. Schumacher, *Phys. Rev. A* **53**, 2046 (1996).
- [30] Z. Zhang, A. Ahmadain, and I. Klich, *Proc. Natl. Acad. Sci. USA* **114**, 5142 (2017).
- [31] B. Mohar, *Linear Alg. Appl.* **103**, 119 (1988).
- [32] F. R. K. Chung and P. Tetali, *Comb. Probab. Comput.* **7**, 141 (1998).
- [33] F. Chung, *Ann. Comb.* **9**, 1 (2005).
- [34] Note that $\tau_{\text{RB}}(G)$ can take on any nonnegative real value. In reality, the creation of a quantum state will always take an integer number of steps greater than or equal to one in our model. Therefore, $\lceil \tau_{\text{RB}} \rceil$ can be used as a measure of the “number of rounds required” in cases where this is important.
- [35] O. Goldreich, in *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, Lecture Notes in Computer Science (Springer, Berlin, 2011), pp. 451–464.
- [36] M. Ajtai, J. Komlós, and E. Szemerédi, in *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC'83)* (ACM, New York, NY, 1983), pp. 1–9.
- [37] O. Reingold, *J. ACM* **55**, 17 (2008).
- [38] I. Dinur, *J. ACM* **54**, 12 (2007).
- [39] S. Arora, S. Rao, and U. Vazirani, *J. ACM* **56** (2009).
- [40] P. Elias, A. Feinstein, and C. Shannon, *IRE Trans. Inf. Theory* **2**, 117 (1956).
- [41] L. R. Ford and D. R. Fulkerson, *Can. J. Math.* **8**, 399 (1956).
- [42] D. R. Fulkerson and Rand Corporation, *Notes on Linear Programming. Part XVI, A Network-Flow Feasibility Theorem and Combinatorial Applications*, ASTIA Document No. AD 156011 (Rand Corporation, Santa Monica, CA, 1958).
- [43] Since it is guaranteed to be integer-valued for graphs with integer-valued edge weights, the flow must in fact be of magnitude $\lceil |F|/\tau_{\text{RB}} \rceil$, but this makes no difference to the argument.
- [44] E. Schoute, L. Mancinska, T. Islam, I. Kerenidis, and S. Wehner, [arXiv:1610.05238](https://arxiv.org/abs/1610.05238).
- [45] A. M. Childs, E. Schoute, and C. M. Unsal, in *Proceedings of the 14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC'19)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 135, edited by W. van Dam and L. Mancinska (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), pp. 3:1–3:24.
- [46] R. Orús and J. I. Latorre, *Phys. Rev. A* **69**, 052308 (2004).
- [47] V. M. Kendon and W. J. Munro, *Quantum Inf. Comput.* **6**, 630 (2006).
- [48] N. Schuch, M. M. Wolf, K. G. H. Vollbrecht, and J. I. Cirac, *New J. Phys.* **10**, 033032 (2008).
- [49] K. Van Acoleyen, M. Mariën, and F. Verstraete, *Phys. Rev. Lett.* **111**, 170501 (2013).
- [50] Z.-X. Gong, M. Foss-Feig, F. G. S. L. Brandão, and A. V. Gorshkov, *Phys. Rev. Lett.* **119**, 050501 (2017).
- [51] M. Van den Nest, *Phys. Rev. Lett.* **110**, 060504 (2013).