An entanglement-based volumetric benchmark for near-term quantum hardware[*]

Kathleen E. Hamilton,[1, 2, †] Nouamane Laanait,[3] Akhil Francis,[4] Sophia E. Economou,[5] George S. Barron,[5] Kübra Yeter-Aydeniz,[6] Titus Morris,[1, 2] Harrison Cooley,[7] Muhun Kang,[8] Alexander F. Kemper,[4] and Raphael Pooser[9, 2]

[1]*Quantum Computational Science Group, Oak Ridge National Laboratory, Oak Ridge, TN, 37831, USA*
[2]*Computer Science and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN, 37831, USA*
[3]*Carelon Digital Platforms, 740 West Peachtree Street NW, Atlanta, GA 30308*
[4]*Department of Physics, North Carolina State University, Raleigh, NC 27695 USA*
[5]*Department of Physics, Virginia Tech, Blacksburg, VA USA*
[6]*Emerging Engineering and Physical Sciences Department,*
The MITRE Corporation, McLean, VA, 22102-7539 USA
[7]*Department of Physics, Georgetown University, Washington, DC, 20007 USA*
[8]*Department of Physics, Cornell University, Ithaca, NY, 14583 USA*
[9]*Quantum Computing and Sensing Group, Oak Ridge National Laboratory, Oak Ridge, TN, 37831, USA*

We introduce a volumetric benchmark for near-term quantum platforms based on the generation and verification of genuine entanglement across n-qubits using graph states and direct stabilizer measurements. Our benchmark evaluates the robustness of multipartite and bipartite n-qubit entanglement with respect to many sources of hardware noise: qubit decoherence, CNOT and swap gate noise, and readout error. We demonstrate our benchmark on multiple superconducting qubit platforms available from IBM (`ibmq_belem`, `ibmq_toronto`, `ibmq_guadalupe` and `ibmq_jakarta`). Subsets of $n < 10$ qubits are used for graph state preparation and stabilizer measurement. Evaluation of genuine and biseparable entanglement witnesses we report observations of 5 qubit genuine entanglement, but robust multipartite entanglement is difficult to generate for $n > 4$ qubits and identify two-qubit gate noise as strongly correlated with the quality of genuine multipartite entanglement.

I. INTRODUCTION

Developing benchmarks for quantum processors is crucial to advance the state of the art in noisy intermediate-scale quantum (NISQ) computing. Meaningful hardware benchmarks need to be applicable to different qubit technologies, and adapt to devices with different qubit layouts and basis gate sets. Application and algorithm metrics need to quantify the complexity of the states that can be generated in large Hilbert spaces, and assist in developing intuition behind the effect of hardware noise and performance. While it is difficult to distill the complex operation of a quantum processor to a single number or a small set of numbers, a good benchmark should accurately capture the relative performance of different processors; it should be as simple as possible, and it should have reasonable scaling as a function of the system size. A comparison of the proposed benchmarks found in the current literature shows the trade-off between scaling and detailed state characterization. Tomographic methods [1, 2] obtain highly detailed representation of quantum states and processes, but scale exponentially. Low-level benchmarks characterize hardware elements (e.g., qubits and gates) but interpolating from low-level to high-level (application) performance may not be straightforward. On the other hand, high-level benchmarks may fail to capture the salient features of the processors.

A unifying framework for hardware characterization is provided by the Volumetric Benchmarks (VBs). IBM introduced quantum volume [3], which benchmarks quantum hardware using randomized circuits. Hardware is assigned a scalar value based on the largest circuit that can be successfully executed, where success is defined by the measurement of heavy output from a random circuit. The randomized circuits used to evaluate quantum volume are characterized as *square*: where the width (number of qubits) and depth (number of computational steps) are equal. This framework was generalized in Ref. [4] to rectangular random circuits (where circuit depth can exceed width). Quantum volume as originally proposed is not clearly related to algorithmic performance or structured entanglement measures, although further refinements have become more closely related to algorithm performance [4].

Quantum entanglement has been shown to be a valuable resource in quantum sensing and metrology [5], quantum communications [6, 7] and quantum error correction [8] when computing with pure quantum states [9]. It is therefore reasonable to design benchmarks based on the ability of a processor to generate entangled states. For example, an interesting–but poten-

tially oversimplified — benchmark is the ability to produce Greenberger–Horne–Zeilinger (GHZ) states [10–13]. While GHZ states have been used as a low-level coherent noise characterization tool [14], they only represent one specific class of entanglement out of the plethora of entangled states of n qubits. Beyond GHZ states, specific cluster states and arbitrary graph states have been proposed as benchmarks of near-term quantum hardware [15, 16]. While a number of studies have used GHZ, cluster state or graph state preparation as a standalone benchmark [17–19], individual state preparation as a benchmark does not generally give a global enough picture of the device's capabilities in quantum information processing. Yet, these observations of multipartite entanglement on contemporary processors do demonstrate that NISQ devices are capable of storing large entangled states, further motivating the need for a standard entanglement-based benchmark [20–23].

The challenge in generalizing and improving entanglement-based benchmarks is that there exist too many entangled states of n qubits. More importantly, the role of entanglement as a computational resource is subtle, especially when generic entangled states are considered [24, 25]. As a result, preparing a small number of arbitrary entangled states does not allow for a meaningful benchmark of hardware capabilities, and may have minimal use in both near-term and long-term quantum computing.
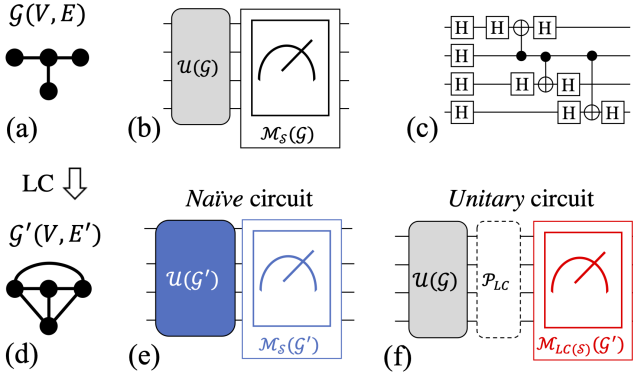

FIG. 1: **Overview of the robust entanglement score** (a) the graph $\mathcal{G}$(V,E) which defines (b) the unitary $\mathcal{U}(\mathcal{G})$ that prepares the graph state $|\psi_{\mathcal{G}}\rangle = |n, E\rangle$ and the stabilizers $\mathcal{M}_{\mathcal{S}}(\mathcal{G})$ for the entanglement witness. (c) The unitary is constructed using Hadamard gates and CNOT gates. (d) Applying a local complement to $\mathcal{G}$(V,E) defines $\mathcal{G}'(V, E')$. (e,f) The associated graph state can be prepared using (e) the *Naïve* method which constructs a new graph unitary $\mathcal{U}(\mathcal{G}')$ using the edge set $E'(\mathcal{G}')$ or (f) the *Unitary* method which appends single qubit operations $\mathcal{P}_{LC}$ to implement the LC operation on the original graph unitary $\mathcal{U}(\mathcal{G})$.

In this work we introduce a VB for entangled state generation, *robust entanglement score* (RES), that en-

compasses practically relevant classes of entangled states and also scales linearly with circuit width. Its intended goal is to show robust genuine entanglement [26] over a large random sampling of entangled quantum states that are used for a variety of quantum information tasks, including quantum error correction, sensing, and one-way computing. At the heart of our new benchmark is the generation and verification of n-qubit entangled *graph* states that are subgraphs of the device connectivity map, and their locally equivalent graph states (Fig. 1(a),(d)). By choosing graph states that are amenable to the device connectivity map, the state preparation circuits can be constructed with minimal two-qubit gates (Fig. 1(c)). But, by generating graph states in the same equivalence class, we test graph states that are not guaranteed to have low gate overhead, or that may require measurement in different bases. Graph states in the same equivalence class can have very different graph structure, from sparse to dense (e.g. containing high degree vertices, triangles).

Entanglement verification is the measure of success for our VB, but successful verification relies on high-fidelity stabilizer measurements that reveal the underlying hardware quality. Thus, through our benchmark we characterize hardware performance that can be used to infer algorithmic performance where the underlying circuit structure is graph-like or for applications that make extensive use of entanglement (e.g. quantum optimization or physics simulations on lattices). Our results show that a benchmark such as RES, which exposes both global and subtle information, and allows for the direct evaluation of application-driven performance, is necessary in capturing and comparing device performance compared to other metrics.

Overall, our benchmark evaluates hardware performance using graph state preparation, independent stabilizer measurements, and quantification of multi-qubit entanglement through the evaluation of genuine and biseparable witnesses [27, 28]. The benchmark leverages specific graph state properties: local complementation of graphs and efficient state characterization via entanglement witnesses [28–30]. This results in global information about the processor performance and its ability to sustain genuine multipartite entanglement among qubit subsets of the processor. These aspects are presented in Section V. We execute our benchmark on superconducting (SC) qubits, but in general our benchmark is hardware-agnostic and can be used to compare different types of devices. Our benchmark is also application agnostic, though graph states are commonly utilized in many quantum applications and act as an algorithmic primitive (see Section III).

Graph States and Entanglement Witnesses

The edges E and vertices V of a simple graph $\mathcal{G}$(V,E) define a unitary that prepares the $n = |V|$ qubit graph state $|\psi_{\mathcal{G}}\rangle = |n, E\rangle = \mathcal{U}(\mathcal{G})|0\rangle^{\otimes n}$ using two-qubit entan-
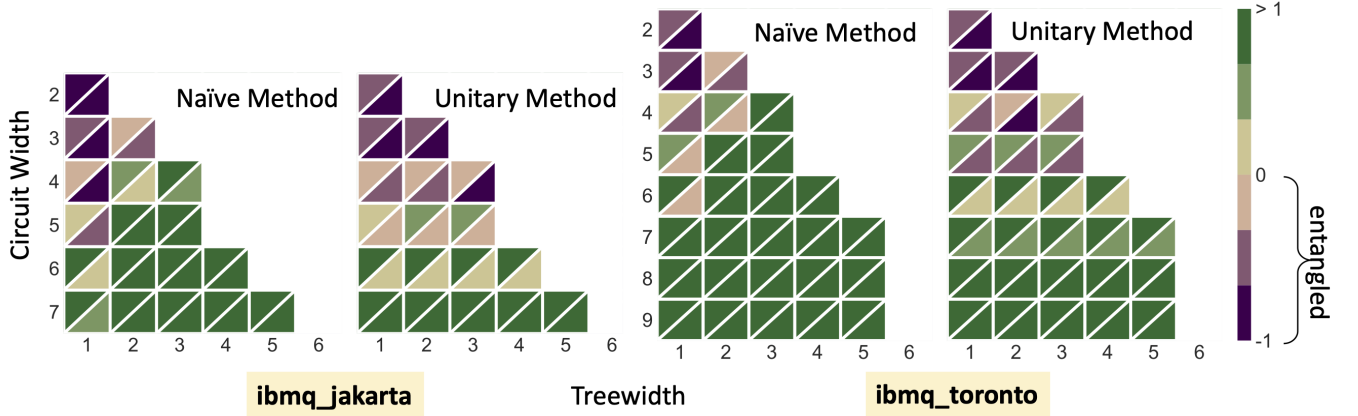
FIG. 2: Median genuine witness values aggregated over multiple graph state executions. The upper triangular values are defined by non-mitigated results, the lower triangular values are defined by error mitigated results, using a tensored mitigator defined in Qiskit `Ignis`. Negative values indicate that entanglement.(Left) Benchmarking of `ibmq_jakarta` (RES-N= 6, RES-U= 12). (Right) Benchmarking of `ibmq_toronto` (RES-N= 6, RES-U= 8).

gling operations specified by the edge set $E(\mathcal{G})$. We verify the generation of multipartite and bipartite entanglement using witness functions $(\mathcal{W})$ constructed from the graph state stabilizers $\mathcal{M}_S(\mathcal{G})$ [27, 28, 30, 31]. Negativity, $\langle \mathcal{W} \rangle < 0$, indicates entanglement (see Section V C).

Robust Entanglement Score

Volumetric benchmarks are defined by finite length features $(\mathbf{w} = (w_0, \ldots, w_i))$ that then define ensembles of circuits $\mathcal{C}(\mathbf{w})$ and a "success criterion" that quantifies performance [4]. The RES benchmark uses the set of states $\{\mathcal{G}'_i\}$ which are equivalent under local operations and classical communication (LOCC) [32]. From an initial n-qubit graph state $\mathcal{G}$ this set of states are generated through sequences of local complement (LC) operations [33–35] (see Section V B). Thus, from $\mathcal{G}$ we use random sequences of LC operations to randomly sample from graph states which are in the same equivalence class. Once the graphs are determined, the corresponding circuits are constructed in two ways: by generating a new graph state circuit [c.f. Fig. 1(e)] (the *Naïve* method) or by appending local unitaries to the state preparation circuit of $\mathcal{G}$ [c.f. Fig. 1(f)] (the *Unitary* method). The *Unitary* method maintains the original number of two-qubit operations, whereas for the *Naïve* method the number of two-qubit operations depends on $E(\mathcal{G}')$.

The features used to evaluate the RES benchmark are the quantum circuit width $(w_0 = n = |V(\mathcal{G})|)$, and graph treewidth $w_1 = \text{tw}(\mathcal{G})$ [36, 37]. Treewidth can be used to identify tree graphs $(\text{tw}(\mathcal{T}) = 1$ independent of $n)$ or complete graphs $(\text{tw}(K_n) = (n-1))$ and is independent of the circuit transpilation. In classical computing, treewidth has been used to quantify problem difficulty [38, 39], computational complexity [40] and complexity of database queries [41].

Success for the RES benchmark is witness function negativity (genuine or biseparable). We assign the RES score for a given backend using the largest set of n-qubits that can be genuinely entangled multiplied by the largest treewidth $(\max w_0 \times \max w_1)$. We distinguish the RES scores by the method used: either the *Naïve* method (RES-Naïve) or the *Unitary* method (RES-Unitary).

II. EVALUATION ON IBM HARDWARE

The RES benchmark was executed on multiple superconducting devices available through IBM Quantum [42], listed in Table I with their respective quantum volumes as reported by IBM. In Figs. 2,3 we present the median genuine witness values evaluated on four IBM backends. Using graph circuits of increasing order we determine what is the largest treewidth, and largest set of qubits (n) that can be genuinely entangled and verified. This is not a test of which qubits on the hardware that can be entangled, but starting from arbitrary qubit subsets, what is the largest entangled state that can be prepared and how robust is it to various sources of hardware noise. The smallest number of qubits that we tested entanglement across was $n = 2$ and the maximum number of qubits was $n = 9$ (or the maximum number of qubits available on a device). All backends were accessed through cloud-based queues, and the data collection method is described in Appendix V D. Tensored mitigators were evaluated during data collection using the Qiskit library `Ignis` [1] [43]. The RES-Naïve and RES-Unitary values reported in Table I are computed from unmitigated values. Mitigated RES values are reported in Table II.

————

[1] Library now depreciated, functionality available in `qiskit-terra` and `qiskit-experiments`
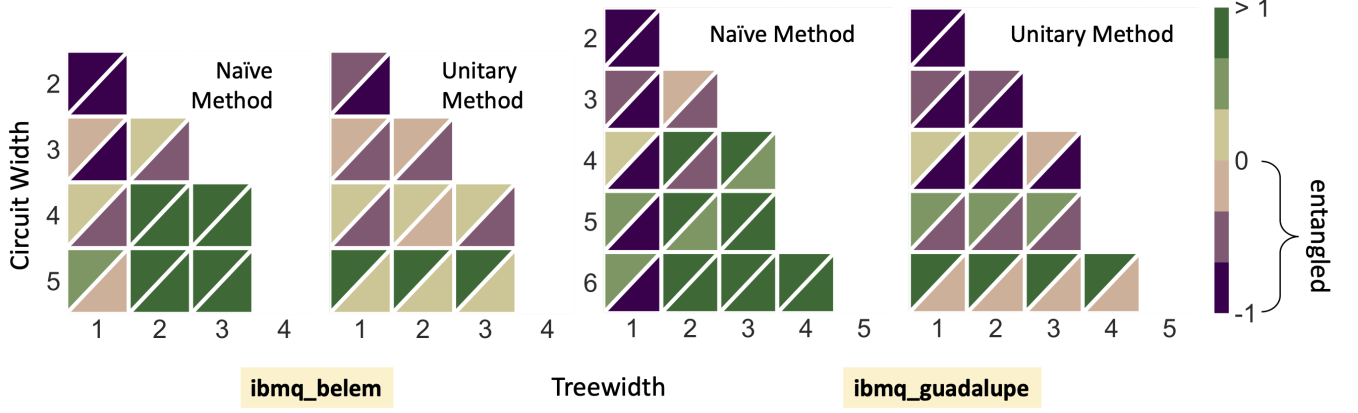
FIG. 3: Median genuine witness values aggregated over multiple graph state executions. The upper triangular values are defined by non-mitigated results, the lower triangular values are defined by error mitigated results, using a tensored mitigator defined in Qiskit `Ignis`. Negative values indicate entanglement. (Left) Benchmarking of `ibmq_belem` (RES-N= 3, RES-U= 6). (Right) Benchmarking of `ibmq_guadalupe` (RES-N= 6, RES-U= 6).

Backend Name	Processor	Qubits	Quantum Volume	RES-Naïve	RES-Unitary
`ibmq_toronto`	r4	27	32	6	8
`ibmq_guadalupe`	r4p	16	32	6	6
`ibmq_jakarta`	r5.11H	7	16	6	12
`ibmq_belem`	r4T	5	16	3	6

TABLE I: Quantum backends, processor types, total number of qubits available, published Quantum Volume, RES-Naïve and RES-Unitary values.

Preparation of genuine entanglement over arbitrary graphs states represents a stringent benchmark, but biseparable entanglement and sub-chain entanglement are also readily measurable on NISQ devices [13]. In Appendix A, we report similar heatmaps for the median biseparable witness results. In general, all backends support high quality bipartite entanglement; the scores are commensurate with other work which has demonstrated multiple two-qubit chains entangled on specific backends.

Overall we observe that in the aggregate it is very difficult to generate and validate genuine n-qubit entanglement that is robust against two-qubit gate noise or circuit depth. On the other hand, robust biseparable entanglement was detectable for qubit sizes up to $n = 9$. Readout error mitigation provides moderate improvement for both the genuine and biseparable witness value.

III. EFFECT OF HARDWARE NOISE ON STABILIZER MEASUREMENT

A. Stabilizer Measurement Fidelity

The color scales of the heat maps shown in Figs. 2,3 are defined by the median genuine entanglement witness values evaluated over multiple graph states randomly drawn from the LC graph orbit of n-qubit hardware graphs. The evaluation of the witness values is dependent on the measurement of individual stabilizers of each graph state, and the ability to generate and detect entanglement in an n-qubit graph state is thus determined by the capability of hardware to prepare and measure these stabilizers with high fidelity. In this section we delve into the individual stabilizer measurements and discuss factors that affect stabilizer measurement fidelity: circuit width, stabilizer weight, circuit scheduling, and treewidth. These features are chosen for our analysis based on the Pearson r-coefficients (see Appendix F for details). As shown in Fig. 4, the median stabilizer measurement computed over all n qubit graph states decreases with increasing graph order, with the exception of $n = 2$ results generated on `ibmq_toronto`. The median expectation values obtained with the *Naïve* method of graph state construction decrease more rapidly with the number of qubits than the *Unitary* method. This performance can likely be attributed to the increased number of CNOTs, since an increase in circuit depth through CNOT additions is likely to cause worse performance than an increase in circuit depth caused by local complement (single qubit) operations in the *Unitary* method as single qubit error rates are lower than the CNOT error rates. In Fig. 4 there are notable outliers in the stabilizer expectation values with $n = 2$ qubits, specifically for data collection on qubit subsets: $[0, 1], [2, 3], [3, 5]$. On `ibmq_toronto` we
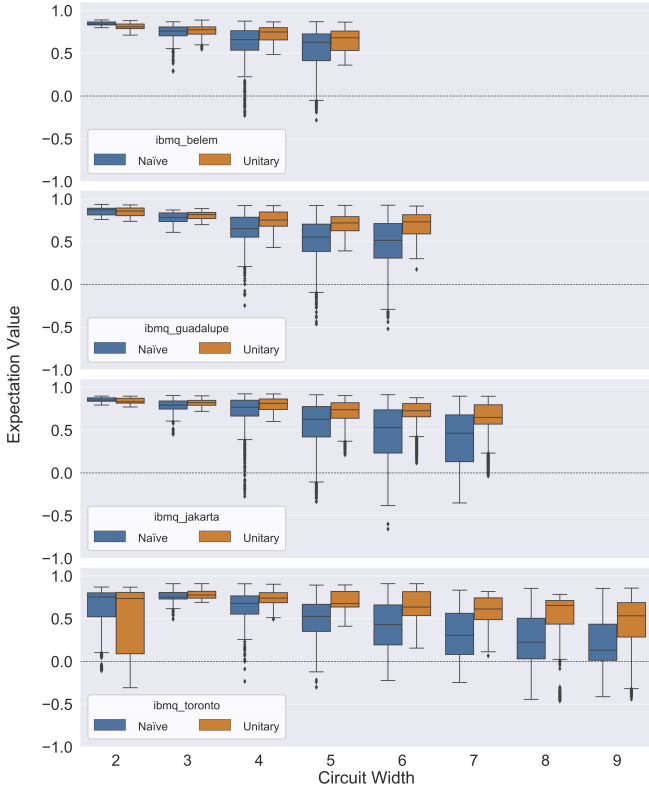
FIG. 4: Quartiles of stabilizer expectation values calculated over all measurements evaluated per circuit width.

observe that the $n = 2$ stabilizer measurements exhibit a large negative skew – we discuss the connection to low level calibration metrics in Appendix D.

B. Error mitigation

Executing the graph state circuits on near-term quantum devices result in noisy measurements. Hardware noise is generated by many sources such as qubit decoherence, gate noise and readout error. We consider how individual stabilizer measurements can be affected by two-qubit gate error, readout error or stabilizer weight (non identity measurements). Figure 16, for example, in Appendix F shows the correlations between these low-level circuit structure datum and expectation values. The Pearson correlation coefficients are computed pairwise between: circuit width, and number of CNOTs in the transpiled circuits, stabilizer weights, graph treewidth and the observed stabilizer measurement.

The gate scheduling is not optimized, but we investigate the effect of readout error mitigation on benchmark performance. For our chosen set of states and backends there are many methods available for noise and error mitigation. For the circuits executed on IBM backends we use the error mitigation library `Ignis` in Qiskit [43] to construct tensored error mitigators: a matrix of noise

characterization that is used to post-process the returned distribution over bitstrings prior to evaluation of individual expectation values. Ref [28] also showed that the inclusion of additional stabilizer measurements improves the robustness of the genuine witness to white noise. Currently, we evaluate these additional terms to test stabilizer measurement stability, but they are not included in the evaluation of the entanglement witness.

In Table I we reported the RES-N and RES-U scores for each backend, based on the unmitigated median witness values. As seen in Figs. 5 and 6, the application of readout error mitigation results in a general shift of stabilizer expectation values closer to 1.0. As seen in Figs. 3, 2 the use of readout error mitigation also improves the median genuine witness value. As a result the RES-N and RES-U scores for each backend are generally improved. We report the effect of readout error mitigation (ROEM) on the RES scores for each backend in Table II. In general increased CNOT counts via SWAP additions

Backend Name	RES Naïve + ROEM	RES Unitary + ROEM
ibmq_toronto	8	15
ibmq_guadalupe	8	24
ibmq_jakarta	6	15
ibmq_belem	6	12

TABLE II: Quantum backends, and mitigated RES values evaluated using tensored mitigators.

lead to expected degraded performance. Exceptions exist when readout noise contains strong correlated noise, the stabilizer expectation values veer far from describing entanglement when particular qubits are used which show large amounts of correlated readout noise. We tested this noise source on SC hardware by examining the effect of readout noise mitigation on the two-qubit subsystem that exhibited the most readout noise on `ibmq_toronto`. We found that highly-correlated qubits on this device drastically reduced the probability of successfully preparing an entangled state.

Figures 5 and 6 show the change in the distribution of stabilizer expectation values for each backend after a tensored noise mitigator was used to implement readout error mitigation. Overall, the effect of noise mitigation is a general shift rightwards towards the ideal value $\langle g_i \rangle = 1$. It is possible that the noise mitigation step will result in non-physical values $\langle g_i \rangle > 1$. When we compute the entanglement witness values with noise mitigated stabilizer measurements we clip the expectation values at $\max(g_i, 1.0)$. A future version of the benchmark will include correlated readout noise mitigation rather than the tensored method in order to connect the observed performance with low-level correlations on-chip. Unless explicitly stated our analysis in the remainder of Section III is of unmitigated measurement data.
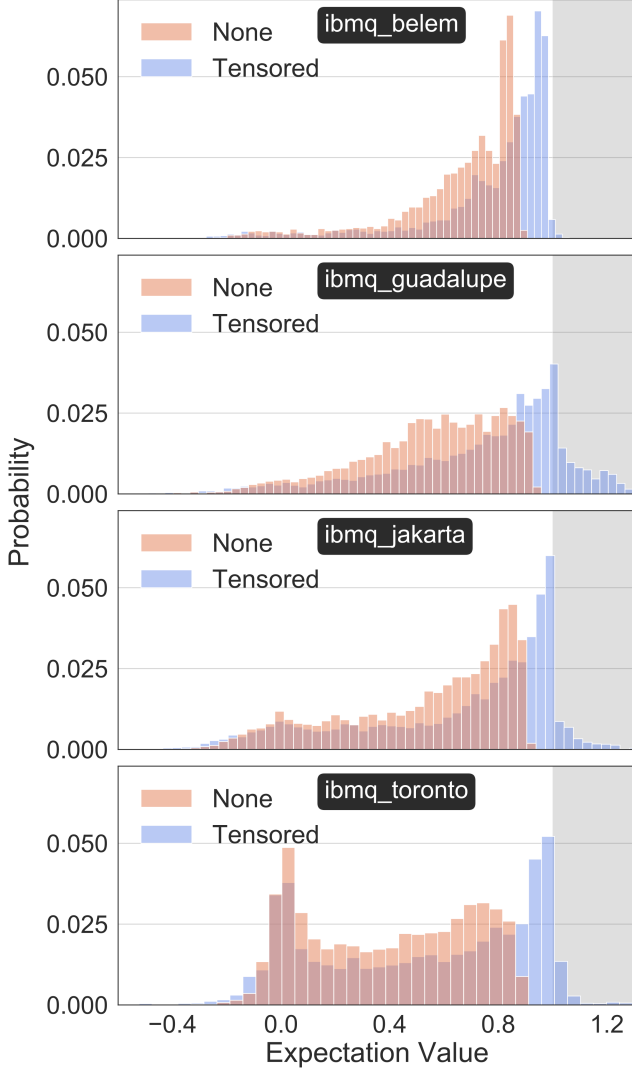
FIG. 5: Distribution of stabilizer values measured on SC qubit backends using the *Naïve* method. Readout error mitigation is implemented using tensored construction of assignment error matrices. Shaded region indicates non-physical values generated by noise mitigation.
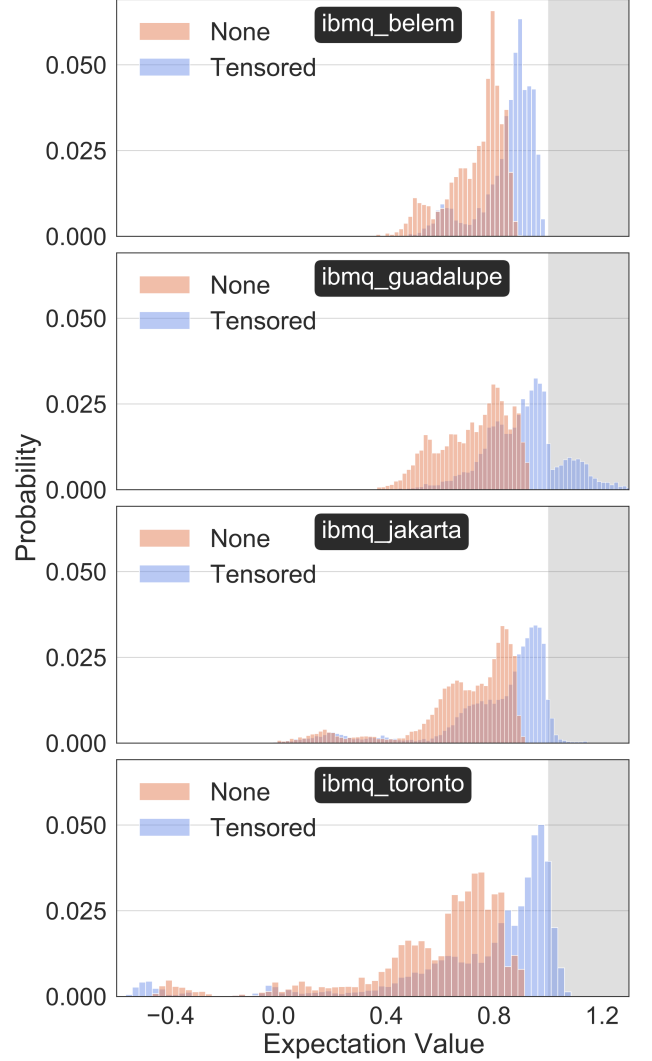

FIG. 6: Distribution of stabilizer values measured on SC qubit backends using the *Unitary* method. Readout error mitigation is implemented using tensored construction of assignment error matrices. Shaded region indicates non-physical values generated by noise mitigation.

IV. CONNECTING BENCHMARK PERFORMANCE TO APPLICATION PERFORMANCE

The focus of this volumetric benchmark is the generation and verification of robust n-qubit genuine (multipartite) entanglement. But in benchmarking entanglement we are interested in exploring the control of long-range interactions and correlations between qubits [44]. In contrast to existing volumetric benchmarks which rely on random circuits, we argue that our benchmark can be closely related to near-term application performance or expanded and adapted to other applications. For example, connections between instantaneous quantum polynomial (IQP) circuits and state preparation for weighted

graph states [45], or planar graph states [46] have been established. We can apply our benchmark to planar and non-planar graph states, the circuit construction can be expanded to include hypergraphs and weighted graphs, although state verification will need to be modified away from stabilizer measurements [32]. In this section we discuss additional areas where benchmark performance can be used to infer future application performance on NISQ hardware.

First, the metric acts as a surrogate model for near-term algorithm and application performance. Many applications executed on NISQ hardware may not map directly into the sparse qubit connectivity. When a densely connected circuit on n qubits is transpiled into a sparsely connected device, the resulting instruction set may have

many additional two-qubit gates. Using ideal graphs, i.e. those that embed into the native hardware graph with minimal overhead, we can estimate a "best case" performance in the presence of hardware noise (gate noise, qubit decoherence and readout error). The non-ideal graph states from the *Unitary* method then test the robustness of n-qubit entanglement to swap gate noise or circuit depth.

Second, the benchmark gathers data through the construction of various graph states, and the measurement of n stabilizers through repeated state preparation. This data can be leveraged to develop noise mitigation methods for either the bitstring distribution or the stabilizer expectation values. Additionally, the variability in the individual stabilizer values can be used to infer the stability of state preparation for variational algorithms [47]. Overall the ability to efficiently characterize how close a prepared state is to a target graph state, without the need for full tomography, makes this benchmark appealing and scalable.

V. METHODS

Graph states provide us a flexible and extensible means of generating entangled states which also provides insight into the hardware capabilities [27, 28, 31, 32]. This section will describe the theory behind entanglement witnesses, the design of the NISQ circuits for witness evaluation and finally how the metric is evaluated.

A. Benchmark circuit construction

A given NISQ device is characterized by the qubit technology, the hardware noise, and the hardware native graph $\mathcal{B}$ which is the fixed layout of couplers and waveguides between physical qubits. This layout defines which two-qubit gates can be implemented with low overhead, or without incurring additional swap gates. For each of the IBM backends we use in this work (`ibmq_belem`, `ibmq_jakarta`, `ibmq_guadalupe`, and `ibmq_toronto`), these layouts are given in Fig. 11 of the Appendix.

The state preparation circuit for $|\psi_{\mathcal{G}}\rangle = |n, E\rangle$ is constructed in two steps [c.f. Fig. 1(c)]. First, the all-zero register $|0\rangle^{\otimes n}$ is transformed into $|+\rangle^{\otimes n}$ by applying Hadamard gates to all qubits in the register. Next, each two-qubit entanglement operation defined by the undirected edge set $E(\mathcal{G})$ is decomposed into a CNOT gate and two Hadamard gates acting on the target qubit following the sequence used in [17]. In this initial demonstration of the benchmark, we limit the degree of optimization in gate scheduling used during the transpilation step. The graph state construction does not schedule gates to reduce gate depth or optimize gate scheduling in general. We apply barriers to prevent the consolidation of the single qubit gates applied in the *Unitary* method. In future demonstrations, different compilation and tran-

spilation techniques could be incorporated to improve the benchmark performance.
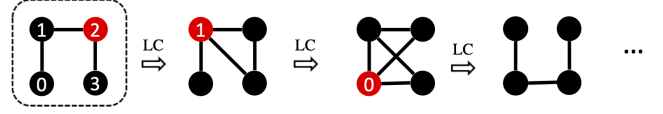
B. LC equivalent state preparation

FIG. 7: The transformation of an initial 4 vertex path graph (far left) using local graph complements. The vertex used to define the local complement at each step is highlighted in red.

Evaluating the entanglement metric samples from a finite set of locally equivalent graph states for an initial graph state. This equivalence class is defined by local complement operations (LC). A local complement of a graph $\mathcal{G}$ is defined by a vertex $v_i \in V(\mathcal{G})$ and its neighborhood $N(v_i)$. The complement of the neighborhood graph $N'(v_i)$ is used to define $\mathcal{G}' = \mathcal{G} \bigcup N'$. An example of LC operation on a 4 vertex path graph is shown in Fig. (7). We describe the two methods for implementing LC operations on graph states that we evaluate in this study: the *Unitary* method and the *Naïve* method considering an initial graph $\mathcal{G}$ and a specific LC operation (ℓ).

The *Unitary* method implements the action of a LC operation (ℓ) on vertex (a) using local single qubit gates [32, 48],

$$\ell_a = e^{-i\frac{\pi}{4}\sigma_x^a} \prod_{b \to a} e^{i\frac{\pi}{4}\sigma_z^b} \qquad (1)$$

where $b \to a$ denotes all neighbors of vertex (a). The graph state circuit is initially constructed with the edge list of $\mathcal{G}$, then the LC operation $(\ell : \mathcal{G} \to \mathcal{G}')$ modifies the circuit $\mathcal{U}_{\mathcal{G}} \to \mathcal{U}'_{\mathcal{G}'}$ by appending X- and- Z axis rotations to $\mathcal{U}_{\mathcal{G}}$ according to Eq. 1. The *Unitary* method also modifies the stabilizer strings $\mathcal{M}_{\mathcal{S}} \to \mathcal{M}_{\mathrm{LC}(\mathcal{S})}$ [32, 49]. This method keeps the number of controlled rotation gates minimal, at the expense of circuit depth and with the requirement of different basis transformations (X, Y, Z).

The *Naïve* method implements a LC operation by wholly redefining $\mathcal{U}_{\mathrm{G}}$. The LC operation is first applied to the underlying graph, $\ell : \mathcal{G} \to \mathcal{G}'$, and the resulting graph $\mathcal{G}'$, is used to construct the graph state circuit. This method of modifying the underlying graph may lead to graph state circuits with two qubit gates that do not embed into $\mathcal{B}$ with minimal overhead, as a result this method may incur additional noise and overhead in the form of swap gates. However, the definition of the stabilizer strings will only using the measurement settings X, Z.

We close this section by noting that each subgraph extracted from the hardware native graphs corresponded to graphs in different LC equivalence classes as shown

in Table V of [32]. This is relevant because while different graphs can be contained as subgraphs of individual graphs– for example the 4-vertex star graph is seen as a subgraph in several graphs of the 4-vertex path graph orbit shown in Fig. 7, the graphs generated in each respective LC orbit were not contained in another LC orbit.

C. Entanglement witnesses for graph states

A state is entangled when it is not fully separable. Multipartite entanglement has richer structure than a bipartite entanglement. A state that is not biseparable is called genuinely multipartite entangled. Biseparable entanglement witnesses and genuine entanglement witnesses can be used to characterize the entanglement. Constructing these witnesses for a general state $|\Psi\rangle$ is difficult, but since we are considering graph states we can leverage the graph state stabilizers.

For each vertex in $v_k \in V(\mathcal{G})$ there is an associated generator constructed from the Pauli Z, X operators,

$$g_k^{(\mathcal{G})} = X^{(k)} \prod_{l \to k} Z^{(l)}, \tag{2}$$

where $l \to k$ denotes the product over all neighbors of vertex (k). For vertices that are not neighbors of (k) an identity operator is inserted. Each n-qubit graph state has a set of $n+1$ stabilizers, the n generators plus the identity string. These Pauli strings are measured and their expectation values are used to compute the entanglement witness functions. These measurements are done serially with multiple preparations of the graph state (see Fig. 1(b)) and measuring each individual stabilizer string.

The genuine entanglement witness operator $\mathcal{W}_\mathrm{G}$ can be constructed using all n generator measurements:

$$\mathcal{W}_\mathrm{G} = (n-1)\mathbb{1} - \sum_k g_k^{\mathcal{G}}. \tag{3}$$

Each genuine entanglement witnesses constructed as in Eq. 3 requires n stabilizer measurements for each n-qubit circuit. This implementation is not optimized for noise robustness: as stated in Ref. [27] the witnesses defined with n stabilizer measurements are only robust against noise levels up to $p_{noise} < 1/n$. Many studies have investigated entanglement witness construction [2, 8, 50–56]. Entanglement witnesses were also recently used in a fault-tolerant weight-4 parity check measurement scheme [57], which demonstrated genuine six-qubit multipartite entanglement in their shuttling-based trapped-ion quantum computer. However, for the purposes of our benchmark, we consider the witness construction in Eqs. 3,A1 to be optimal because it uses the same number of stabilizer measurements per graph, and the entanglement of each generated graph state is evaluated using witnesses with the same level of noise robustness.

D. Data Collection

Given n (number of qubits), we can induce a subgraph of n vertices on a backend with hardware native graph $\mathcal{G}_\mathcal{B}$ (see Fig. 11 in Appendix B). This subgraph can be used to prepare a specific graph state, and we sample a set of LC equivalent graph states from its LC graph orbit. This is done by generating random sequences of LC operations: $[a, b, c]$ corresponds to applying the LC at vertex (a), followed by the LC at vertex (b), followed by the LC at vertex (c). Depending on the induced subgraph, the LC graph orbit can be very small (e.g. see the LC graph orbits associated with GHZ states in Appendix B) or it can contain a large number of graphs (see Table V of Ref. [32]). For n qubits we generate a minimum of 2^{n+1} sequence of LC sequences. Each sequence length is drawn uniformly at random from $[1, 2n]$ and the only constraints applied to the random sequences is that duplicate operations (e.g. $[a, a]$) are consolidated to $[a]$. We do not require that unique graphs are generated by each sequence, it is possible that two sequences $\ell = [a, b, c]$ and $\ell' = [a', b', c']$ will generate the same graph. It is also possible that a sequence will have no net effect on the induced subgraph of the hardware native graph.

Once the LC sequences have been generated, each individual graph state preparation circuit is constructed in Qiskit n times to evaluate the n stabilizers of the target graph state, plus the identity string (measurement in the computational basis only). Overall, a sequence of L LC sequence of a n qubit graph state will require nL circuits. This data is collected using large job batches. Each backend available via the IBM Q Experience has a maximum number of experiments that can be sent under the same job batch. We generate circuit sets (see pseudocode in Appendix C) which are partitioned into job batches (each graph, each set of stabilizers) such that the maximum number of experiments are sent to the backend. However, if multiple batches are needed, then the jobs are partitioned such that all circuits for a specific graph state are sent in the same batch. After a batch has been sent to a backend and executed, we send the necessary circuits to evaluate the tensored mitigator.

VI. DATA AVAILABILITY

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

VII. CONCLUSIONS

The benchmark we have introduced in this paper has applications and benefits to near-term, NISQ computing as well as longer term applications. It is challenging to develop metrics that capture the impact of errors, but including additional states of the Hilbert space would give

a more thorough characterization of the device. In the near term, the data collected during the execution of this study has amassed a large corpus of state preparation results. This resource will be leveraged to advance near-term noise mitigation methods that can be applied to the 2^n distribution of bitstring counts or the individual expectation values. Also in the near term, our benchmark can be used to infer performance on variational quantum algorithms (VQAs). Our benchmark relies on Clifford circuits, and to demonstrate the potential benefits for VQAs we point to recent studies that have leveraged Clifford circuits for bootstrapped initialization of VQAs [58]. In this work, Clifford circuits are used to find regions of the optimization landscape to facilitate training. The specific graph state preparations used in our benchmark can capture the dynamics and impact of errors in different regions of the Hilbert space, and in different bases.

Adaptable, informative benchmarking will assist in the development of hardware and algorithms that reach this milestone. While the generation of n-qubit multi-partite entanglement may not be a final goal of near-term computing, quantifying this entanglement can be connected with many applications and serves as an effective way to benchmark quantum hardware.

COMPETING INTERESTS

The Authors declare no Competing Financial or Non-Financial Interests

AUTHOR CONTRIBUTIONS

K.E.H., N. L., S. E., A. K., A. F. , K.Y.A., and R. C. P. contributed to the scientific design.
K.E.H., A. K., A. F. , K.Y.A., and R. C. P. contributed to the data collection.
N. L., K.E.H., G. B., A.F., A.K., T. M., H. C. and M. K. contributed to the software development and testing. K.E.H., A. F., A.K., R.C.P. , contributed to the data analysis and interpretation.
All authors contributed to the manuscript writing and editing.

ACKNOWLEDGEMENTS

[1] Shelby Kimmel, Marcus P da Silva, Colm A Ryan, Blake R Johnson, and Thomas Ohki. Robust extraction of tomographic information via randomized benchmarking. *Physical Review X*, 4(1):011050, 2014.

[2] Mordecai Waegell and Justin Dressel. Benchmarks of nonclassicality for qubit arrays. *npj Quantum Information*, 5(1):1–8, 2019.

[3] Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3):032328, 2019.

[4] Robin Blume-Kohout and Kevin C Young. A volumetric framework for quantum computer benchmarks. *Quantum*, 4:362, 2020.

[5] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Advances in quantum metrology. *Nature photonics*, 5(4):222–229, 2011.

[6] A Pirker, J Wallnöfer, and W Dür. Modular architectures for quantum networks. *New Journal of Physics*, 20(5):053054, 2018.

[7] Michael Epping, Hermann Kampermann, Dagmar Bruß, et al. Multi-partite entanglement can speed up quantum key distribution in networks. *New Journal of Physics*, 19(9):093012, 2017.

[8] Andrea Rodriguez-Blanco, Alejandro Bermudez, Markus Müller, and Farid Shahandeh. Efficient and robust certification of genuine multipartite entanglement in noisy quantum error correction circuits. *PRX Quantum*, 2:020304, Apr 2021.

[9] Richard Jozsa and Noah Linden. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 459(2036):2011–2032, 2003.

[10] Otfried Gühne, Chao-Yang Lu, Wei-Bo Gao, and Jian-Wei Pan. Toolbox for entanglement detection and fidelity estimation. *Physical Review A*, 76(3):030305, 2007.

[11] Tom Mainiero. Homological Tools for the Quantum Mechanic. *arXiv:1901.02011 [hep-th, physics:quant-ph]*, January 2019. arXiv: 1901.02011.

[12] Ken X Wei, Isaac Lauer, Srikanth Srinivasan, Neereja Sundaresan, Douglas T McClure, David Toyli, David C McKay, Jay M Gambetta, and Sarah Sheldon. Verifying multipartite entangled greenberger-horne-zeilinger states via multiple quantum coherences. *Physical Review A*, 101(3):032343, 2020.

[13] Gary J Mooney, Gregory AL White, Charles D Hill, and Lloyd CL Hollenberg. Generation and verification of 27-qubit greenberger-horne-zeilinger states in a superconducting quantum computer. *Journal of Physics Communications*, 5(9):095004, 2021.

[14] You Zhou. Entanglement detection under coherent noise: Greenberger-horne-zeilinger-like states. *Physical Review A*, 101(1):012301, 2020.

[15] Thomas Nutz, Antony Milne, Pete Shadbolt, and Terry Rudolph. Proposal for demonstration of long-range cluster state entanglement in the presence of photon loss. *APL Photonics*, 2(6):066103, 2017.

[16] You Zhou, Qi Zhao, Xiao Yuan, and Xiongfeng Ma. Detecting multipartite entanglement structure with minimal resources. *npj Quantum Information*, 5(1):1–8, 2019.

[17] Yuanhao Wang, Ying Li, Zhang-qi Yin, and Bei Zeng. 16-qubit IBM universal quantum computer can be fully entangled. *npj Quantum information*, 4(1):1–6, 2018.

[18] Gary J Mooney, Charles D Hill, and Lloyd CL Hollenberg. Entanglement in a 20-qubit superconducting quantum computer. *Scientific reports*, 9(1):1–8, 2019.

[19] Gary J. Mooney, Gregory A. L. White, Charles D. Hill, and Lloyd C. L. Hollenberg. Whole-device entanglement in a 65-qubit superconducting quantum computer. *arXiv:2102.11521 [quant-ph]*, February 2021. arXiv: 2102.11521.

[20] Joab Morais Varela, Ranieri Nery, George Moreno, Alice Caroline de Oliveira Viana, Gabriel Landi, and Rafael Chaves. Enhancing entanglement and total correlation dynamics via local unitaries. *Physical Review A*, 105(2):022430, 2022.

[21] Zachary Eldredge, Leo Zhou, Aniruddha Bapat, James R Garrison, Abhinav Deshpande, Frederic T Chong, and Alexey V Gorshkov. Entanglement bounds on the performance of quantum computing architectures. *Physical Review Research*, 2(3):033316, 2020.

[22] Yanzhu Chen, Linghua Zhu, Chenxu Liu, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. How much entanglement do quantum optimization algorithms require? *arXiv preprint arXiv:2205.12283*, 2022.

[23] Maxime Dupont, Nicolas Didier, Mark J Hodson, Joel E Moore, and Matthew J Reagor. An entanglement perspective on the quantum approximate optimization algorithm. *arXiv preprint arXiv:2206.07024*, 2022.

[24] David Gross, Steve T Flammia, and Jens Eisert. Most quantum states are too entangled to be useful as computational resources. *Physical Review Letters*, 102(19):190501, 2009.

[25] Michael J Bremner, Caterina Mora, and Andreas Winter. Are random pure states useful for quantum computation? *Physical Review Letters*, 102(19):190502, 2009.

[26] Guifré Vidal and Rolf Tarrach. Robustness of entanglement. *Physical Review A*, 59(1):141, 1999.

[27] Géza Tóth and Otfried Gühne. Detecting genuine multipartite entanglement with two local measurements. *Physical Review Letters*, 94(6):060501, 2005.

[28] Bastian Jungnitsch, Tobias Moroder, and Otfried Gühne. Entanglement witnesses for graph states: General theory and examples. *Physical Review A*, 84(3):032310, 2011.

[29] Mohamed Bourennane, Manfred Eibl, Christian Kurtsiefer, Sascha Gaertner, Harald Weinfurter, Otfried Gühne, Philipp Hyllus, Dagmar Bruß, Maciej Lewenstein, and Anna Sanpera. Experimental detection of multipartite entanglement using witness operators. *Physical Review Letters*, 92(8):087902, 2004.

[30] Bastian Jungnitsch, Tobias Moroder, and Otfried Gühne. Taming multiparticle entanglement. *Physical Review Letters*, 106(19):190502, 2011.

[31] Géza Tóth and Otfried Gühne. Entanglement detection in the stabilizer formalism. *Physical Review A*, 72(2):022340, 2005.

[32] Marc Hein, Wolfgang Dür, Jens Eisert, Robert Raussendorf, M Nest, and H-J Briegel. Entanglement in graph states and its applications. *arXiv preprint quant-ph/0602096*, 2006.

[33] André Bouchet. An efficient algorithm to recognize locally equivalent graphs. *Combinatorica*, 11(4):315–329, 1991.

[34] André Bouchet. Recognizing locally equivalent graphs. *Discrete Mathematics*, 114(1-3):75–86, 1993.

[35] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Efficient algorithm to recognize the local clifford equivalence of graph states. *Physical Review A*, 70(3):034302, 2004.

[36] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

[37] Hans L Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[38] Erik D Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 637–646. IEEE, 2005.

[39] Dániel Marx. Can you beat treewidth? In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 169–179. IEEE, 2007.

[40] Eric Goles, Pedro Montealegre, Martín Ríos Wilson, and Guillaume Theyssier. On the impact of treewidth in the computational complexity of freezing dynamics. In *Conference on Computability in Europe*, pages 260–272. Springer, 2021.

[41] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1), mar 2007.

[42] IBM quantum. https://quantum-computing.ibm.com/. 2022.

[43] MD SAJID ANIS, Abby-Mitchell, Héctor Abraham, AduOffei, Rochisha Agarwal, Gabriele Agliardi, Merav Aharoni, Vishnu Ajith, Ismail Yunus Akhalwaya, Gadi Aleksandrowicz, Thomas Alexander, Matthew Amy, Sashwat Anagolum, Anthony-Gandon, Eli Arbel, Abraham Asfaw, Anish Athalye, Artur Avkhadiev, Carlos Azaustre, PRATHAMESH BHOLE, Abhik Banerjee, Santanu Banerjee, Will Bang, Aman Bansal, Panagiotis

Barkoutsos, Ashish Barnawal, George Barron, George S. Barron, Luciano Bello, Yael Ben-Haim, M. Chandler Bennett, Daniel Bevenius, Dhruv Bhatnagar, Prakhar Bhatnagar, Arjun Bhobe, Paolo Bianchini, Lev S. Bishop, Carsten Blank, Sorin Bolos, Soham Bopardikar, Samuel Bosch, Sebastian Brandhofer, Brandon, Sergey Bravyi, Nick Bronn, Bryce-Fuller, David Bucher, Artemiy Burov, Fran Cabrera, Padraic Calpin, Lauren Capelluto, Jorge Carballo, Ginés Carrascal, Adam Carriker, Ivan Carvalho, Adrian Chen, Chun-Fu Chen, Edward Chen, Jielun (Chris) Chen, Richard Chen, Franck Chevallier, Kartik Chinda, Rathish Cholarajan, Jerry M. Chow, Spencer Churchill, CisterMoke, Christian Claus, Christian Clauss, Caleb Clothier, Romilly Cocking, Ryan Cocuzzo, Jordan Connor, Filipe Correa, Zachary Crockett, Abigail J. Cross, Andrew W. Cross, Simon Cross, Juan Cruz-Benito, Chris Culver, Antonio D. Córcoles-Gonzales, Navaneeth D, Sean Dague, Tareq El Dandachi, Animesh N Dangwal, Jonathan Daniel, Marcus Daniels, Matthieu Dartiailh, Abdón Rodríguez Davila, Faisal Debouni, Anton Dekusar, Amol Deshmukh, Mohit Deshpande, Delton Ding, Jun Doi, Eli M. Dow, Patrick Downing, Eric Drechsler, Eugene Dumitrescu, Karel Dumon, Ivan Duran, Kareem EL-Safty, Eric Eastman, Grant Eberle, Amir Ebrahimi, Pieter Eendebak, Daniel Egger, ElePT, Emilio, Alberto Espiricueta, Mark Everitt, Davide Facoetti, Farida, Paco Martín Fernández, Samuele Ferracin, Davide Ferrari, Axel Hernández Ferrera, Romain Fouilland, Albert Frisch, Andreas Fuhrer, Bryce Fuller, MELVIN GEORGE, Julien Gacon, Borja Godoy Gago, Claudio Gambella, Jay M. Gambetta, Adhisha Gammanpila, Luis Garcia, Tanya Garg, Shelly Garion, James R. Garrison, Jim Garrison, Tim Gates, Hristo Georgiev, Leron Gil, Austin Gilliam, Aditya Giridharan, Glen, Juan Gomez-Mosquera, Gonzalo, Salvador de la Puente González, Jesse Gorzinski, Ian Gould, Donny Greenberg, Dmitry Grinko, Wen Guan, Dani Guijo, John A. Gunnels, Harshit Gupta, Naman Gupta, Jakob M. Günther, Mikael Haglund, Isabel Haide, Ikko Hamamura, Omar Costa Hamido, Frank Harkins, Kevin Hartman, Areeq Hasan, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Stefan Hillmich, Hiroshi Horii, Connor Howington, Shaohan Hu, Wei Hu, Chih-Han Huang, Junye Huang, Rolf Huisman, Haruki Imai, Takashi Imamichi, Kazuaki Ishizaki, Ishwor, Raban Iten, Toshinari Itoko, Alexander Ivrii, Ali Javadi, Ali Javadi-Abhari, Wahaj Javed, Qian Jianhua, Madhav Jivrajani, Kiran Johns, Scott Johnstun, Jonathan-Shoemaker, JosDenmark, JoshDumo, John Judge, Tal Kachmann, Akshay Kale, Naoki Kanazawa, Jessica Kane, Kang-Bae, Annanay Kapila, Anton Karazeev, Paul Kassebaum, Tobias Kehrer, Josh Kelso, Scott Kelso, Hugo van Kemenade, Vismai Khanderao, Spencer King, Yuri Kobayashi, Kovi11Day, Arseny Kovyrshin, Rajiv Krishnakumar, Pradeep Krishnamurthy, Vivek Krishnan, Kevin Krsulich, Prasad Kumkar, Gawel Kus, Ryan LaRose, Enrique Lacal, Raphaël Lambert, Haggai Landa, John Lapeyre, Joe Latone, Scott Lawrence, Christina Lee, Gushu Li, Tan Jun Liang, Jake Lishman, Dennis Liu, Peng Liu, Lolcroc, Abhishek K M, Liam Madden, Yunho Maeng, Saurav Maheshkar, Kahan Majmudar, Aleksei Malyshev, Mohamed El Mandouh, Joshua Manela, Manjula, Jakub Marecek, Manoel Marques, Kunal Marwaha, Dmitri Maslov, Paweł Maszota, Dolph

Mathews, Atsushi Matsuo, Farai Mazhandu, Doug McClure, Maureen McElaney, Cameron McGarry, David McKay, Dan McPherson, Srujan Meesala, Dekel Meirom, Corey Mendell, Thomas Metcalfe, Martin Mevissen, Andrew Meyer, Antonio Mezzacapo, Rohit Midha, Daniel Miller, Hannah Miller, Zlatko Minev, Abby Mitchell, Nikolaj Moll, Alejandro Montanez, Gabriel Monteiro, Michael Duane Mooring, Renier Morales, Niall Moran, David Morcuende, Seif Mostafa, Mario Motta, Romain Moyard, Prakash Murali, Daiki Murata, Jan Müggenburg, Tristan NEMOZ, David Nadlinger, Ken Nakanishi, Giacomo Nannicini, Paul Nation, Edwin Navarro, Yehuda Naveh, Scott Wyman Neagle, Patrick Neuweiler, Aziz Ngoueya, Thien Nguyen, Johan Nicander, Nick-Singstock, Pradeep Niroula, Hassi Norlen, NuoWenLei, Lee James O'Riordan, Oluwatobi Ogunbayo, Pauline Ollitrault, Tamiya Onodera, Raul Otaolea, Steven Oud, Dan Padilha, Hanhee Paik, Soham Pal, Yuchen Pang, Ashish Panigrahi, Vincent R. Pascuzzi, Simone Perriello, Eric Peterson, Anna Phan, Kuba Pilch, Francesco Piro, Marco Pistoia, Christophe Piveteau, Julia Plewa, Pierre Pocreau, Alejandro Pozas-Kerstjens, Rafał Pracht, Milos Prokop, Viktor Prutyanov, Sumit Puri, Daniel Puzzuoli, Pythonix, Jesús Pérez, Quant02, Quintiii, Rafey Iqbal Rahman, Arun Raja, Roshan Rajeev, Isha Rajput, Nipun Ramagiri, Anirudh Rao, Rudy Raymond, Oliver Reardon-Smith, Rafael Martín-Cuevas Redondo, Max Reuter, Julia Rice, Matt Riedemann, Rietesh, Drew Risinger, Pedro Rivero, Marcello La Rocca, Diego M. Rodríguez, RohithKarur, Ben Rosand, Max Rossmannek, Mingi Ryu, Tharrmashastha SAPV, Nahum Rosa Cruz Sa, Arijit Saha, Abdullah Ash-Saki, Sankalp Sanand, Martin Sandberg, Hirmay Sandesara, Ritvik Sapra, Hayk Sargsyan, Aniruddha Sarkar, Ninad Sathaye, Niko Savola, Bruno Schmitt, Chris Schnabel, Zachary Schoenfeld, Travis L. Scholten, Eddie Schoute, Mark Schulterbrandt, Joachim Schwarm, James Seaward, Sergi, Ismael Faro Sertage, Kanav Setia, Freya Shah, Nathan Shammah, Will Shanks, Rohan Sharma, Yunong Shi, Jonathan Shoemaker, Adenilton Silva, Andrea Simonetto, Deeksha Singh, Divyanshu Singh, Parmeet Singh, Phattharaporn Singkanipa, Yukio Siraichi, Siri, Jesús Sistos, Iskandar Sitdikov, Seyon Sivarajah, Slavikmew, Magnus Berg Sletfjerding, John A. Smolin, Mathias Soeken, Igor Olegovich Sokolov, Igor Sokolov, Vicente P. Soloviev, SooluThomas, Starfish, Dominik Steenken, Matt Stypulkoski, Adrien Suau, Shaojun Sun, Kevin J. Sung, Makoto Suwama, Oskar Słowik, Hitomi Takahashi, Tanvesh Takawale, Ivano Tavernelli, Charles Taylor, Pete Taylour, Soolu Thomas, Kevin Tian, Mathieu Tillet, Maddy Tod, Miroslav Tomasik, Caroline Tornow, Enrique de la Torre, Juan Luis Sánchez Toural, Kenso Trabing, Matthew Treinish, Dimitar Trenev, TrishaPe, Felix Truger, Georgios Tsilimigkounakis, Davindra Tulsi, Doğukan Tuna, Wes Turner, Yotam Vaknin, Carmen Recio Valcarce, Francois Varchon, Adish Vartak, Almudena Carrera Vazquez, Prajjwal Vijaywargiya, Victor Villar, Bhargav Vishnu, Desiree Vogt-Lee, Christophe Vuillot, James Weaver, Johannes Weidenfeller, Rafal Wieczorek, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, WinterSoldier, Jack J. Woehr, Stefan Woerner, Ryan Woo, Christopher J. Wood, Ryan Wood, Steve Wood, James Wootton, Matt Wright, Lucy Xing, Jintao YU, Bo Yang, Unchun Yang, Jimmy

Yao, Daniyar Yeralin, Ryota Yonekura, David Yonge-Mallo, Ryuhei Yoshida, Richard Young, Jessie Yu, Lebin Yu, Yuma-Nakamura, Christopher Zachow, Laura Zdanski, Helena Zhang, Iulia Zidaru, Bastian Zimmermann, Christa Zoufal, aeddins ibm, alexzhang13, b63, bartek bartlomiej, bcamorrison, brandhsn, chetmurthy, deeplokhande, dekel.meirom, dime10, dlasecki, ehchen, ewinston, fanizzamarco, fs1132429, gadial, galeinston, georgezhou20, georgios ts, gruu, hhorii, hhyap, hykavitha, itoko, jeppevinkel, jessica angel7, jezerjojo14, jliu45, johannesgreiner, jscott2, klinvill, krutik2966, ma5x, michelle4654, msuwama, nico lgrs, nrhawkins, ntgiwsvp, ordmoj, sagar pahwa, pritamsinha2304, rithikaadiga, ryancocuzzo, saktar unr, saswati qiskit, septembrr, sethmerkel, sg495, shaashwat, smturro2, sternparky, strickroman, tigerjack, tsura crisaldo, upsideon, vadebayo49, welien, willhbang, wmurphy collabstar, yang.luh, and Mantas Čepulkovskis. Qiskit: An open-source framework for quantum computing, 2021.

[44] Alexander Streltsov, Gerardo Adesso, and Martin B Plenio. Colloquium: Quantum coherence as a resource. *Reviews of Modern Physics*, 89(4):041003, 2017.

[45] Masahito Hayashi and Yuki Takeuchi. Verifying commuting quantum computations via fidelity estimation of weighted graph states. *New Journal of Physics*, 21(9):093060, 2019.

[46] Keisuke Fujii and Tomoyuki Morimae. Commuting quantum circuits and complexity of ising partition functions. *New Journal of Physics*, 19(3):033003, 2017.

[47] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[48] Adán Cabello, Lars Eirik Danielsen, Antonio J. López-Tarrida, and José R. Portillo. Optimal preparation of graph states. *Phys. Rev. A*, 83:042314, Apr 2011.

[49] Adán Cabello, Antonio J López-Tarrida, Pilar Moreno, and José R Portillo. Compact set of invariants characterizing graph states of up to eight qubits. *Physical Review A*, 80(1):012102, 2009.

[50] Dariusz Chruściński and Gniewomir Sarbicki. Entanglement witnesses: construction, analysis and classification. *Journal of Physics A: Mathematical and Theoretical*, 47(48):483001, 2014.

[51] Philipp Hauke, Markus Heyl, Luca Tagliacozzo, and Peter Zoller. Measuring multipartite entanglement through dynamic susceptibilities. *Nature Physics*, 12(8):778–782, 2016.

[52] Kyung Hoon Han and Seung-Hyeok Kye. Construction of multi-qubit optimal genuine entanglement witnesses. *Journal of Physics A: Mathematical and Theoretical*, 49(17):175303, 2016.

[53] Farid Shahandeh, Martin Ringbauer, Juan C Loredo, and Timothy C Ralph. Ultrafine entanglement witnessing. *Physical Review Letters*, 118(11):110502, 2017.

[54] David Amaro and Markus Müller. Design and experimental performance of local entanglement witness operators. *Physical Review A*, 101(1):012317, 2020.

[55] Bas Dirkse, Matteo Pompili, Ronald Hanson, Michael Walter, and Stephanie Wehner. Witnessing entanglement in experiments with correlated noise. *Quantum Science and Technology*, 5(3):035007, 2020.

[56] Jan Roik, Karol Bartkiewicz, Antonín Černoch, and Karel Lemr. Accuracy of entanglement detection via artificial neural networks and human-designed entanglement witnesses. *Physical Review Applied*, 15(5):054006, 2021.

[57] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. G. Poschinger. Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer. *Phys. Rev. X*, 12:011032, Feb 2022.

[58] Gokul Subramanian Ravi, Pranav Gokhale, Yi Ding, William M Kirby, Kaitlin N Smith, Jonathan M Baker, Peter J Love, Henry Hoffmann, Kenneth R Brown, and Frederic T Chong. Cafqa: Clifford ansatz for quantum accuracy. *arXiv preprint arXiv:2202.12924*, 2022.
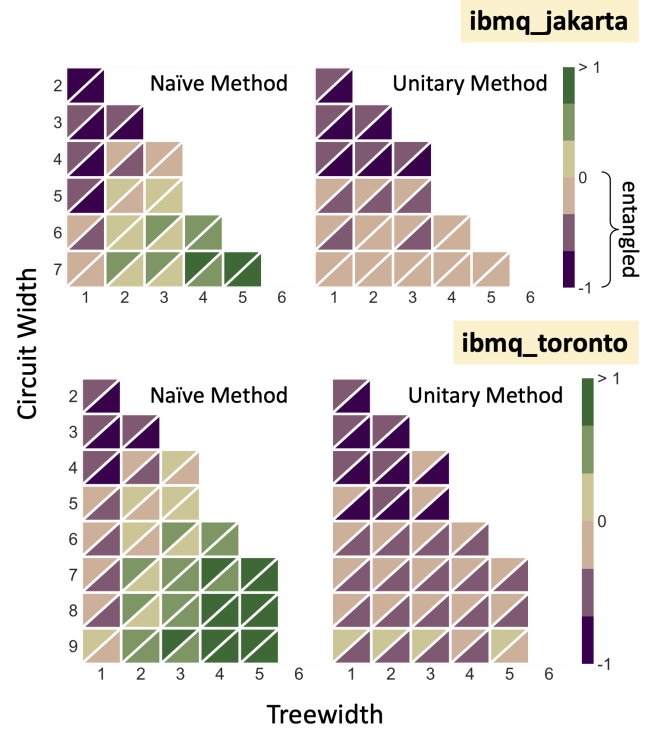
Appendix A: Biseparable Witnesses

FIG. 8: Median biseparable witness values aggregated over multiple graph state executions. The upper triangular values are defined by non-mitigated results, the lower triangular values are defined by error mitigated results, using a tensored mitigator defined in Qiskit Ignis. Negative values indicate that entanglement can be generated and detected. (Top) Volumetric benchmarking of `ibmq_jakarta` (Bottom) `ibmq_toronto` using biseparable entanglement witnesses.

Evaluating the biseparable entanglement witnesses quantifies the separability between pairs of qubits in the n-qubit state. A biseparable entanglement witness can be constructed for each edge $e_{ij} \in E(\mathcal{G})$ and only requires

two measurements (see [31])

$$W_{i,j}(G) = \mathbb{1} - g_i^{(\mathcal{G})} - g_j^{(\mathcal{G})}. \qquad \text{(A1)}$$

In Figures 8, 9 we similarly plot the median value aggregated over all biseparable witness values evaluated from the LC graph orbits. The biseparable witnesses are evaluated using the same set of stabilizer measurements collected from graphs sampled from the LC graph orbit.
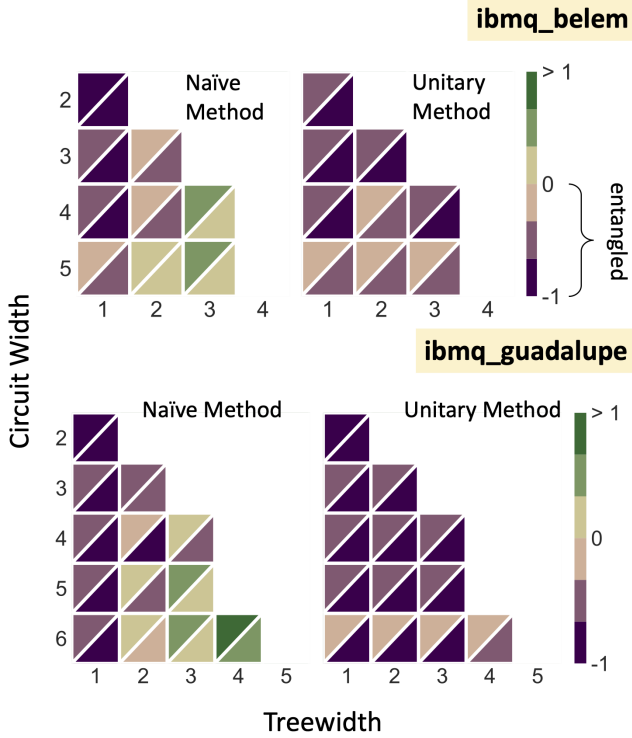

FIG. 9: Median biseparable witness values aggregated over multiple graph state executions. The upper triangular values are defined by non-mitigated results, the lower triangular values are defined by error mitigated results, using a tensored mitigator defined in Qiskit Ignis. Negative values indicate that entanglement can be generated and detected. (Top) Volumetric benchmarking of `ibmq_belem` (Bottom) `ibmq_guadalupe` using biseparable entanglement witnesses.

Appendix B: GHZ state preparation

We have evaluated the robustness of entanglement generated on four IBM machines using our benchmark. Each backend has a specific hardware native graph which defines the possible set of subgraphs that can be extracted on n qubits. By design, our benchmark only considers the graph orbits associated with induced subgraphs of the hardware native graphs defined for a specific backend. In this section we highlight the optimal preparation of n-
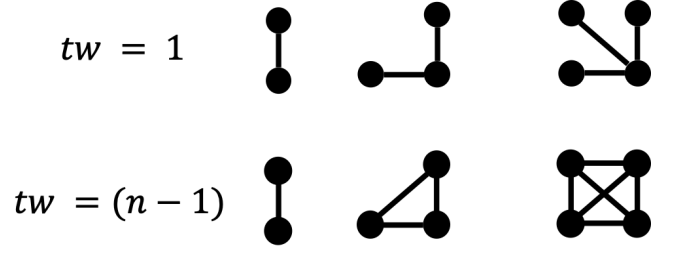

FIG. 10: The graph orbits associated with the $|\text{GHZ}_2\rangle, |\text{GHZ}_3\rangle, |\text{GHZ}_4\rangle$ states that can be prepared from subgraphs directly extracted from the 4 IBM backends we have discusssed in this work. The graphs in each orbit have treewidths of either 1 or $n-1$.

qubit GHZ state $|\text{GHZ}_n\rangle$ encountered during the evaluation of our metric. For the qubit layouts shown in Fig. 11, each backend can prepare the $|\text{GHZ}_2\rangle, |\text{GHZ}_3\rangle, |\text{GHZ}_4\rangle$ graph states from hardware subgraphs (see Fig. 10). The LC graph orbit associated with a n-vertex star graph contains only the star graph ($tw = 1$) and the n-vertex complete graph ($tw = n-1$) (see Fig. 10).

In Tables III and IV we highlight the lowest observed genuine entanglement witnesses and the lowest observed biseparable entanglemnt witnesses for GHZ graph states and their orbits.

Genuine Witness-GHZ orbit				
Name	Qubits	Star	Complete	Method
ibmq_belem	2	-0.740234	-0.740234	*Naïve*
	2	-0.728271	-0.728271	*Unitary*
	3	-0.458252	-0.116211	*Naïve*
	3	-0.541260	-0.458252	*Unitary*
	4	-0.127197	1.232422	*Naïve*
	4	-0.135742	-0.130615	*Unitary*
ibmq_toronto	2	-0.718018	-0.718018	*Naïve*
	2	-0.710693	-0.710693	*Unitary*
	3	-0.476318	-0.185791	*Naïve*
	3	-0.471680	-0.459717	*Unitary*
	4	0.143799	1.496826	*Naïve*
	4	0.142578	0.180176	*Unitary*
ibmq_guadalupe	2	-0.860352	-0.860352	*Naïve*
	2	-0.856201	-0.856201	*Unitary*
	3	-0.568359	-0.191162	*Naïve*
	3	-0.587158	-0.565186	*Unitary*
	4	-0.250244	0.914551	*Naïve*
	4	-0.104980	-0.032227	*Unitary*
ibmq_jakarta	2	-0.792969	-0.792969	*Naïve*
	2	-0.779785	-0.779785	*Unitary*
	3	-0.640381	-0.321045	*Naïve*
	3	-0.552246	-0.540039	*Unitary*
	4	-0.447998	0.727783	*Naïve*
	4	-0.422363	-0.377197	*Unitary*

TABLE III: Minimum non-mitigated genuine witness values measured for n qubit GHZ states prepared on `ibmq_belem`, `ibmq_toronto`, `ibmq_guadalupe`, and `ibmq_jakarta`.
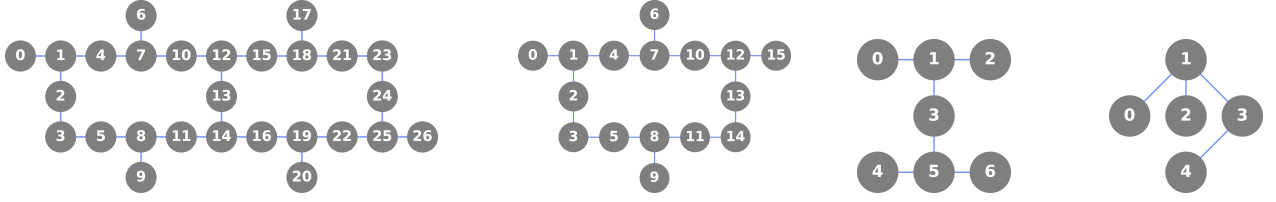
FIG. 11: (Left to Right) Hardware native graphs of `ibmq_toronto`, `ibmq_guadalupe`, `ibmq_jakarta`, and `ibmq_belem`. On `ibmq_toronto`, the largest qubit subset used in this study contained 9 qubits. On `ibmq_guadalupe`, the largest qubit subset used in this study contained 6 qubits. On `ibmq_jakarta` and `ibmq_belem` graph states were generated which used all qubits on the backend.

Biseparable Witness-GHZ orbit				
Name	Qubits	Star	Complete	Method
`ibmq_belem`	2	-0.740234	-0.740234	*Naïve*
	2	-0.728271	-0.728271	*Unitary*
	3	-0.644531	-0.524170	*Naïve*
	3	-0.705566	-0.628662	*Unitary*
	4	-0.583984	-0.225342	*Naïve*
	4	-0.540283	-0.531494	*Unitary*
`ibmq_toronto`	2	-0.718018	-0.718018	*Naïve*
	2	-0.710693	-0.710693	*Unitary*
	3	-0.668213	-0.511719	*Naïve*
	3	-0.667725	-0.663574	*Unitary*
	4	-0.411133	-0.002686	*Naïve*
	4	-0.408691	-0.400391	*Unitary*
`ibmq_guadalupe`	2	-0.860352	-0.860352	*Naïve*
	2	-0.856201	-0.856201	*Unitary*
	3	-0.716309	-0.534180	*Naïve*
	3	-0.729248	-0.708740	*Unitary*
	4	-0.580322	-0.275391	*Naïve*
	4	-0.496826	-0.443604	*Unitary*
`ibmq_jakarta`	2	-0.792969	-0.792969	*Naïve*
	2	-0.779785	-0.779785	*Unitary*
	3	-0.754395	-0.599365	*Naïve*
	3	-0.699707	-0.687988	*Unitary*
	4	-0.710693	-0.425781	*Naïve*
	4	-0.696289	-0.652100	*Unitary*

TABLE IV: Minimum non-mitigated bisepable witness values measured for n qubit GHZ states prepared on `ibmq_belem`, `ibmq_toronto`, `ibmq_guadalupe`, and `ibmq_jakarta` via the *Unitary* method.

Algorithm1 Circuit construction and data collection for the *Naïve* method.

Require: $|\{q_i\}| \geq 2$ ▷ fixed set of hardware qubits
Ensure: $\mathcal{B}[\{q_i\}]$ is connected
  $\mathcal{G} \leftarrow \mathcal{B}[\{q_i\}]$
  $\{\ell\} \leftarrow []$
  for $i \leftarrow 0$ to 2^{n+1} **do** ▷ Generate random LC sequences
    $m \leftarrow x \sim \text{Unif}(0, 2n)$
    $\vec{\ell}_i \leftarrow []$
    for $j \leftarrow 0$ to m **do**
      $\vec{\ell}_i.\text{append}(y \sim \text{Unif}(0, n-1))$
    end for
    $\ell.\text{append}(\vec{\ell}_i)$
  end for
  $\text{M} \leftarrow []$
  for $i \leftarrow 0$ to m **do**
    $\mathcal{G}' \leftarrow \mathcal{G}$
    for $j \leftarrow 0$ to $|\vec{\ell}_i|$ **do**
      $\mathcal{G}' \leftarrow \ell_i[j](\mathcal{G}')$ ▷ Construct LC transformed graph
    end for
    $|\psi_{\mathcal{G}'}\rangle \leftarrow \mathcal{U}(\mathcal{G})|0\rangle^{\otimes n}$ ▷ Prepare state
    $\{g'_k\} \leftarrow g_k^{(\mathcal{G}')}$ ▷ define generators (cf. Eq. 2)
    for $j \leftarrow 0$ to k **do**
      $\text{M}.\text{append}(\langle \psi_{\mathcal{G}'}|g'_j|\psi_{\mathcal{G}'}\rangle)$
    end for
  end for

Appendix C: Pseudocode

We provide pseudocode to implement data collection using the *Naïve* method (in Algorithm 1) and the *Unitary* method (in Algorithm 2) described in the main text.

Algorithm2 Circuit construction and data collection for the *Unitary* method.

Require: $|\{q_i\}| \geq 2$ ▷ fixed set of hardware qubits
Ensure: $\mathcal{B}[\{q_i\}]$ is connected
  $\mathcal{G} \leftarrow \mathcal{B}[\{q_i\}]$
  $\{\ell\} \leftarrow []$
  for $i \leftarrow 0$ to 2^{n+1} **do**   ▷ Generate random LC sequences
    $m \leftarrow x \sim \text{Unif}(0, 2n)$
    $\vec{\ell}_i \leftarrow []$
    for $j \leftarrow 0$ to m **do**
      $\vec{\ell}_i$.append($y \sim \text{Unif}(0, n-1)$)
    end for
    ℓ.append($\vec{\ell}_i$)
  end for
  M $\leftarrow []$
  for $i \leftarrow 0$ to m **do**
    $|\psi_{\mathcal{G}}\rangle \leftarrow \mathcal{U}(\mathcal{G})|0\rangle^{\otimes n}$ ▷ Prepare state
    $\mathcal{U}_S \leftarrow \mathbb{1}$
    for $j \leftarrow 0$ to $|\vec{\ell}_i|$ **do**
      $\mathcal{U}_S$.append($\mathcal{U}(\ell_i[j])$)   ▷ Construct local unitary (cf. Eq. 1)
      $\{g_k\} \leftarrow \{g'_k\}$ ▷ Transform generators (see Ref. [49])
    end for
    M.append($\langle\psi_{\mathcal{G}'}|\mathcal{U}_S^\dagger g'_j \mathcal{U}_S|\psi_{\mathcal{G}'}\rangle$)
  end for

Appendix D: Hardware calibration data

Once the circuit batches are composed (see Section V D), they are sent to cloud-based queues. Prior to sending the job batches, we pull the most recent calibration data available for our target backend, and store this data. This data includes single qubit calibration data (T_1, T_2 values, anharmonicity, frequency) gate calibration data (length, frequency) and also error characterization (single qubit readout error and individual gate error). Based on the features we analyzed in the main text, the efficacy of readout error mitigation and the correlation with number of CNOT gates, we present in Figs. 12,13 the values of the single qubit readout error and CNOT gate error reported by IBM for each backend, collected at the time of data collection.

In Fig. 4 of the main text, the two-qubit stabilizer measurements on `ibmq_toronto` had a large negative skew. We highlight the connection between the qubit subsets that resulted in low stabilizer measurements and the low-level calibration data extracted here. As a threshold we isolate the stabilizer measurements that are below the noise robust threshold of the genuine entanglement witness (1/2). This data was collected in April of 2022.

For this subset of data, the couplers which correspond to low stabilizer measurements were $[0,1], [2,3], [3,5]$. The recorded CNOT gate errors for couplers $[0,1], [1,0], [2,3], [3,2]$ was 1.0, but for couplers $[3,5], [5,3]$ the gate error was reported as 0.024. For the individual qubits, the reported readout errors were: 0.353 for qubit 0, 0.04 for qubit 1, 0.02 for qubit 2, 0.04 for qubit 3, and 0.02 for qubit 5. While the high gate

FIG. 12: Single qubit readout error values collected from calibration data for each backend. Only the error values for the individual hardware qubits used in our data collection are plotted. Calibration data was pulled over several days.
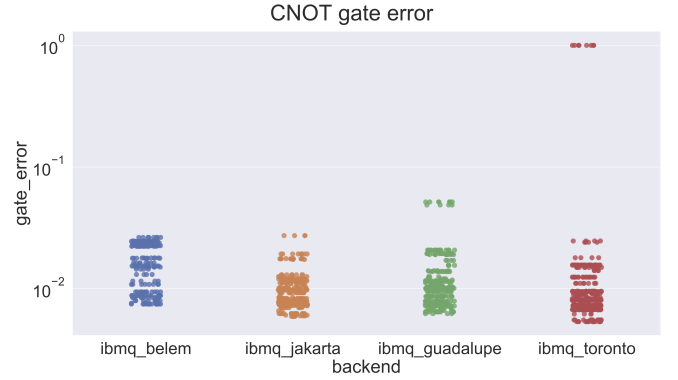

FIG. 13: CNOT gate error values collected from calibration data for each backend plotted on log scale. Only the error values for gates on the individual hardware qubits used in our data collection are plotted. Calibration data was pulled over several days.

error associated with couplers $[0,1], [1,0], [2,3], [3,2]$, and high readout error for qubit 0 may have contributed to the poor preparation of a two-qubit entangled state (Bell state) on hardware qubits 0 and 1, the calibration data for qubits 3, 5 and the associated couplers $[3,5], [5,3]$ does not correlate with the observed low stabilizer measurements.

Appendix E: Minimum Entanglement Witness Values

Figures 8, 9 in the main text show the median witness value evaluated over several graph states drawn from the relative LC graph orbit of n qubit subgraphs. In Table V we report the lowest non-mitigated genuine entanglement witness values measured for n qubits on each backend. Then, in Table VI we report the lowest biseparable witness values evaluated on each backend, using the *Naïve* or the *Unitary* method. When reporting the hardware

qubit subset, we emphasize the pair of qubits used to define the generators used in the specific witness.

Appendix F: Correlation Matrices

In the main text Section III we structure our analysis along several circuit structure features: circuit width, number of in the transpiled circuits, graph treewidth, and stabilizer weight. These features were chosen based on the observed correlations with stabilizer expectation values. For each feature, we compute the Pearson r-coefficient, and show correlation matrix plots for each backend in Figs. 14,15,16,17. All r-coefficient values were found with with $p < 0.001$.

Genuine Witness				
Name	Qubits	*Unitary*	Qubits	*Naïve*
ibmq_belem	[0,1]	-0.72827	[0,1]	-0.740234
	[0, 1, 2]	-0.541260	[0, 1, 2]	-0.458252
	[0, 1,2, 3]	-0.135742	[0, 1,2, 3]	-0.127197
	[0, 1, 2, 3, 4]	0.371826	[0, 1, 2, 3, 4]	0.333252
ibmq_toronto	[1, 2]	-0.710693	[1, 2]	-0.718017
	[5, 8, 11]	-0.471680	[5, 8, 11]	-0.476318
	[5, 8, 11, 14]	-0.143555	[5, 8, 11, 14]	-0.186523
	[3, 5, 8,11, 14]	0.241211	[3, 5, 8,11, 14]	0.275879
	[3, 5, 8,11, 14, 16]	0.659424	[3, 5, 8,11, 14, 16]	0.584717
	[3, 5, 8, 11, 14, 16, 19]	1.558837	[3, 5, 8, 11, 14, 16, 19]	1.526367
	[0, 1, 2, 4, 7, 6, 10, 12]	2.520996	[3, 5, 8, 11,14, 16, 19, 22]	2.414062
	[3, 5, 8, 11, 14, 16, 19, 22, 25]	3.303223	[3, 5, 8, 11, 14, 16, 19, 22, 25]	3.167969
ibmq_guadalupe	[3, 5]	-0.85620	[3, 5]	-0.86035
	[0, 1, 2]	-0.587158	[0, 1, 2]	-0.568359
	[10, 12, 6, 7]	-0.495117	[10, 12, 15, 13]	-0.513427
	[7, 10, 12, 13, 14]	0.070801	[7, 10, 12,13, 14]	0.099609
	[5, 8, 11, 12, 13, 14]	0.267822	[0, 1, 2,4, 6, 7]	0.102051
ibmq_jakarta	[1, 3]	-0.779785	[1, 3]	-0.792969
	[0, 1, 2]	-0.552246	[0, 1, 2]	-0.640381
	[0, 1, 3, 5]	-0.429688	[0, 1, 2, 3]	-0.447998
	[0, 1, 2, 3, 5]	-0.028564	[0, 1, 2,3, 5]	-0.198242
	[0, 1, 2, 3, 5, 6]	0.47729	[0, 1, 2, 3, 5, 6]	0.186279
	[0, 1, 2, 3,5, 4, 6]	1.264648	[0, 1, 2, 3, 5, 4, 6]	1.053223

TABLE V: Hardware qubits that returned the minimum non-mitigated genuine witness values measured on ibmq_belem, ibmq_toronto, ibmq_guadalupe, and ibmq_jakarta.

Biseparable Witness				
Name	Qubits	*Unitary*	Qubits	*Naïve*
ibmq_belem	**[0,1]**	-0.72827	**[0,1]**	-0.74023
	[0, **1,2**]	-0.7056	[0, **1,2**]	-0.6445
	[2, 1, **3,4**]	-0.5991	[2, 1, **3,4**]	-0.6057
	[0, 1, 2, **3,4**]	-0.5630	[0, 1, 2, **3,4**]	-0.6086
ibmq_toronto	**[1,2]**	-0.710693	**[1,2]**	-0.718017
	[5, **8,11**]	-0.667725	[5, **8,11**]	-0.668213
	[5, 8, **11,14**]	-0.709717	[5, 8, **11,14**]	-0.697510
	[3, 5, 8, **11,14**]	-0.737793	[3, 5, 8, **11,14**]	-0.737549
	[3, 5, 8, 11, **14,16**]	-0.747559	[3, 5, 8, 11, **14,16**]	-0.754639
	[3, 5, 8, **11,14**, 16, 19]	-0.600342	[3, 5, 8, 11, 14, **16,19**]	-0.632324
	[3, 5, 8, 11, **14,16**, 19, 22]	-0.501953	[3, 5, 8, **11,14**, 16, 19, 22]	-0.506348
	[3, 5, 8, 11, 14, 16, 19, **22,25**]	-0.645264	[3, 5, 8, 11, 14, 16, 19, **22,25**]	-0.635742
ibmq_guadalupe	**[3,5]**	-0.856201	**[3,5]**	-0.860352
	[0,1, 2]	-0.729248	**[0,1**, 2]	-0.716309
	[11, **12,13**, 14]	-0.771729	[11, **12,13**, 14]	-0.786621
	[7, 10, 12, **13,14**]	-0.730957	[7, 10, 12, **13,14**]	-0.736816
	[5, 8, 11, **12,13**, 14]	-0.779297	[0, 1, 2, 4, **6,7**]	-0.793213
ibmq_jakarta	**[1,3]**	-0.779785	**[1,3]**	-0.792969
	[0, **1,2**]	-0.699707	[0, **1,2**]	-0.754395
	[0, 1, **3,5**]	-0.759277	[0, 1, **3,5**]	-0.762695
	[0, 1, 2, **3,5**]	-0.719971	[0, 1, 2, **3,5**]	-0.753174
	[0, 1, 2, 3, **5,6**]	-0.707031	[0, 1, 2, 3, **5,6**]	-0.749023
	[0, 1, 2, 3, **5,4**, 6]	-0.597900	[0, 1, 2, 3, **5,4**, 6]	-0.5551758

TABLE VI: Minimum non-mitigated biseparable witness values measured for n qubits on ibmq_belem, ibmq_toronto, ibmq_guadalupe, and ibmq_jakarta.
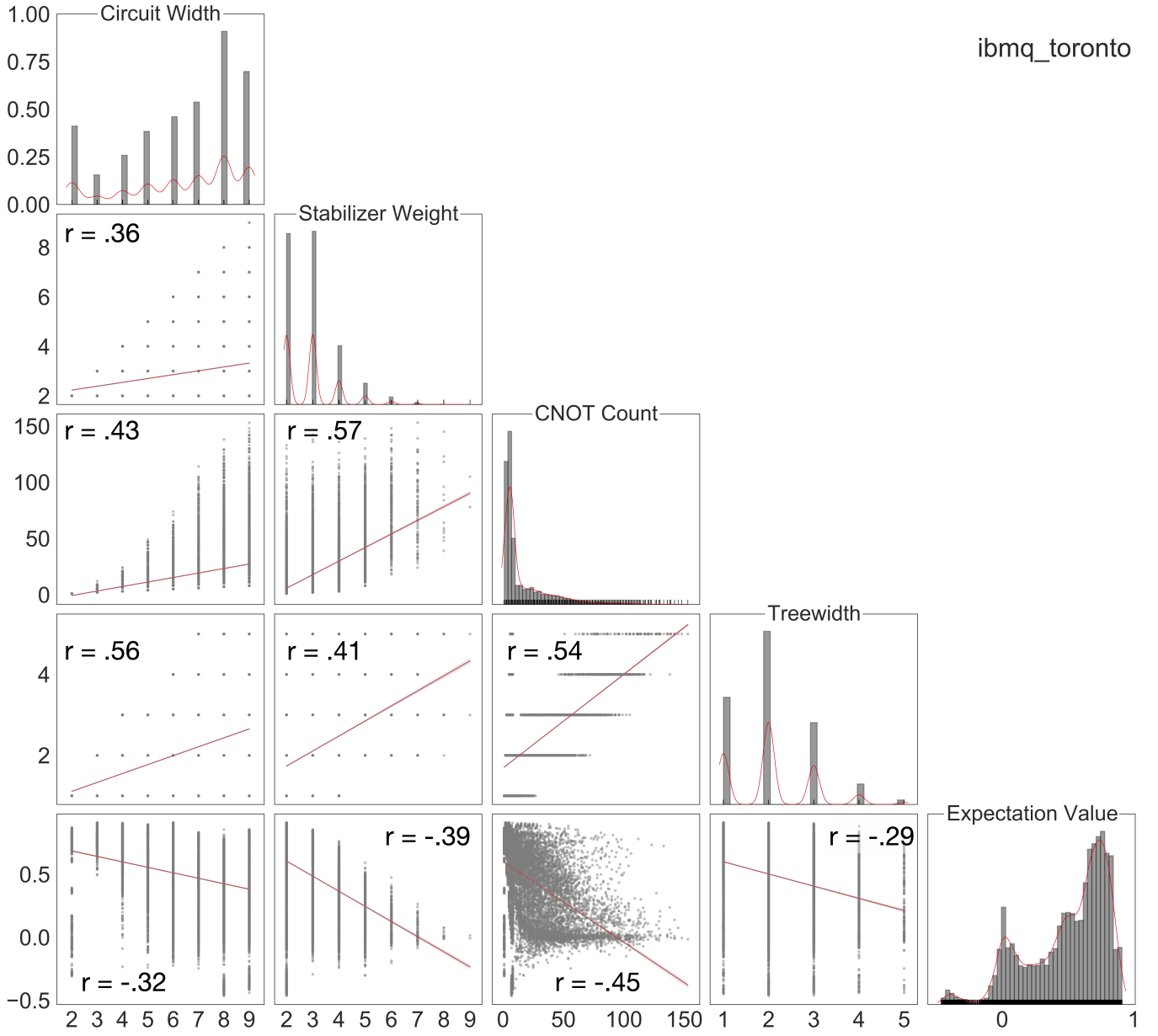
FIG. 14: Correlation matrix for `ibmq_toronto`. Linear regression fit plotted in red with confidence interval and Pearson's r-coefficient computed between features are reported on individual scatter plots. `ibmq_toronto` r-coefficients are computed with 19272 degrees of freedom.
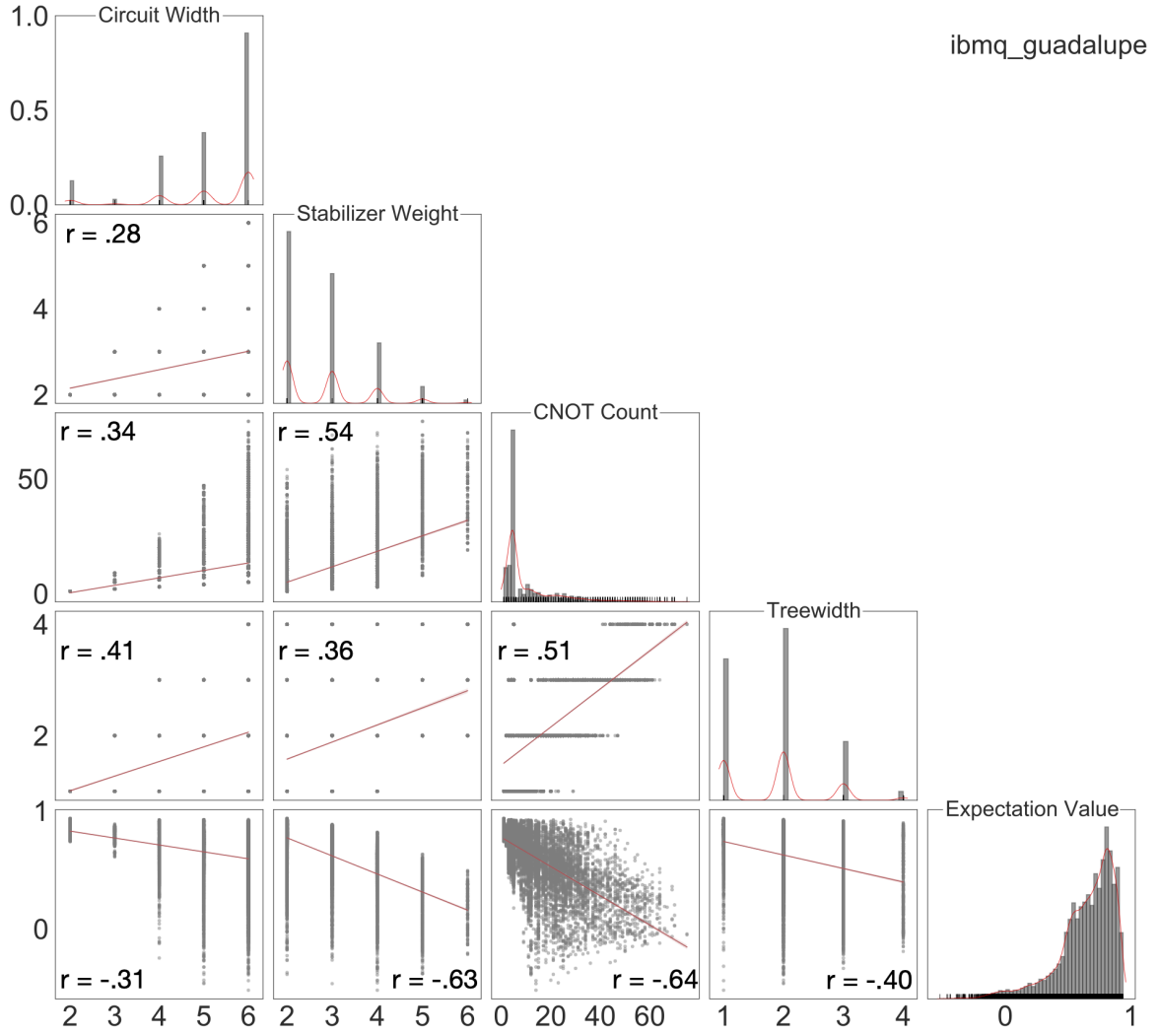
FIG. 15: Correlation matrix for `ibmq_guadalupe`. Linear regression fit plotted in red with confidence interval and Pearson's r-coefficient computed between features are reported on individual scatter plots. `ibmq_guadalupe` r-coefficients are computed with 17368 degrees of freedom.
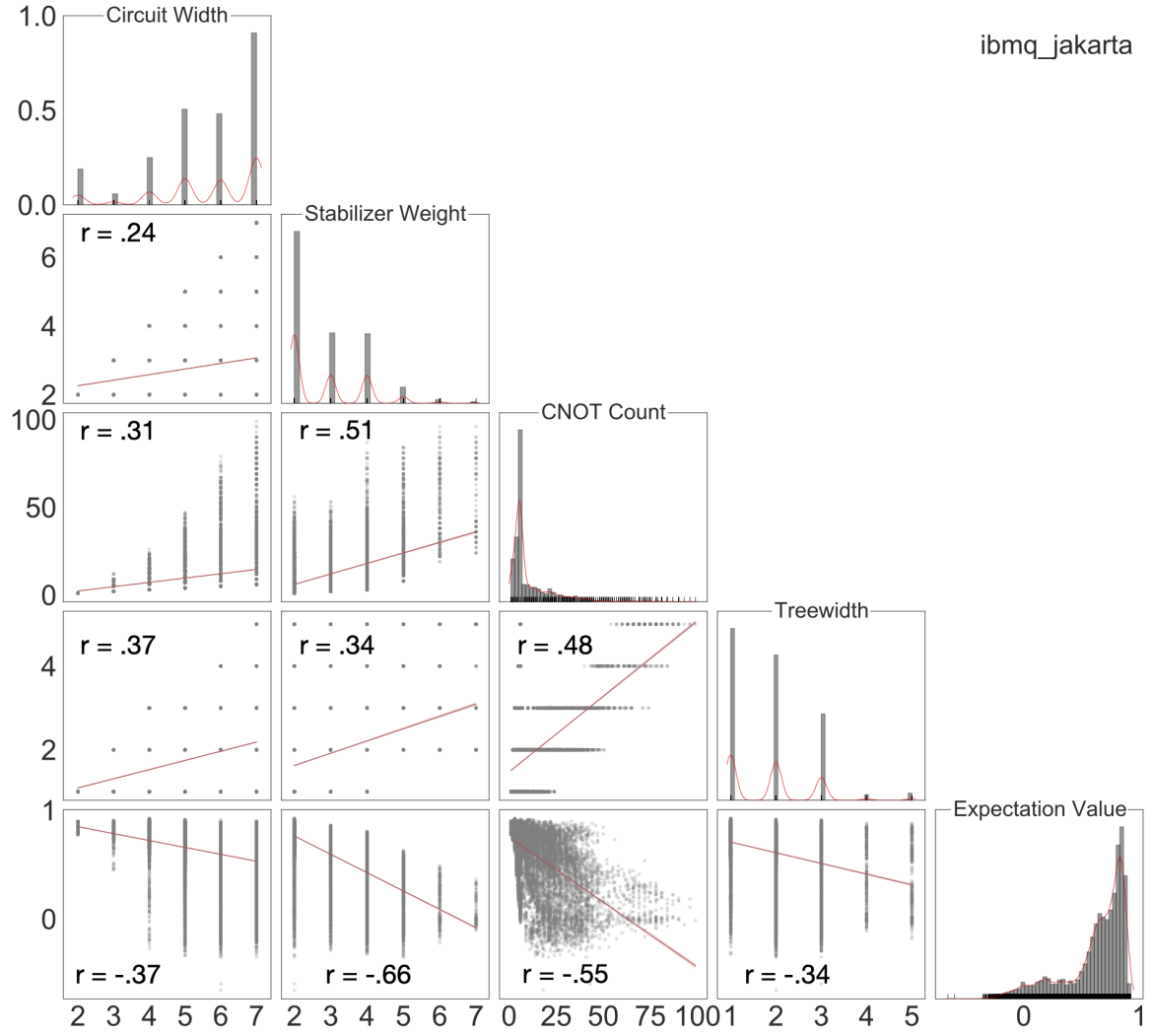
FIG. 16: Correlation matrix for `ibmq_jakarta`. Linear regression fit plotted in red with confidence interval and Pearson's r-coefficient computed between features are reported on individual scatter plots. `ibmq_jakarta` r-coefficient values were computed with 23018 degrees of freedom.
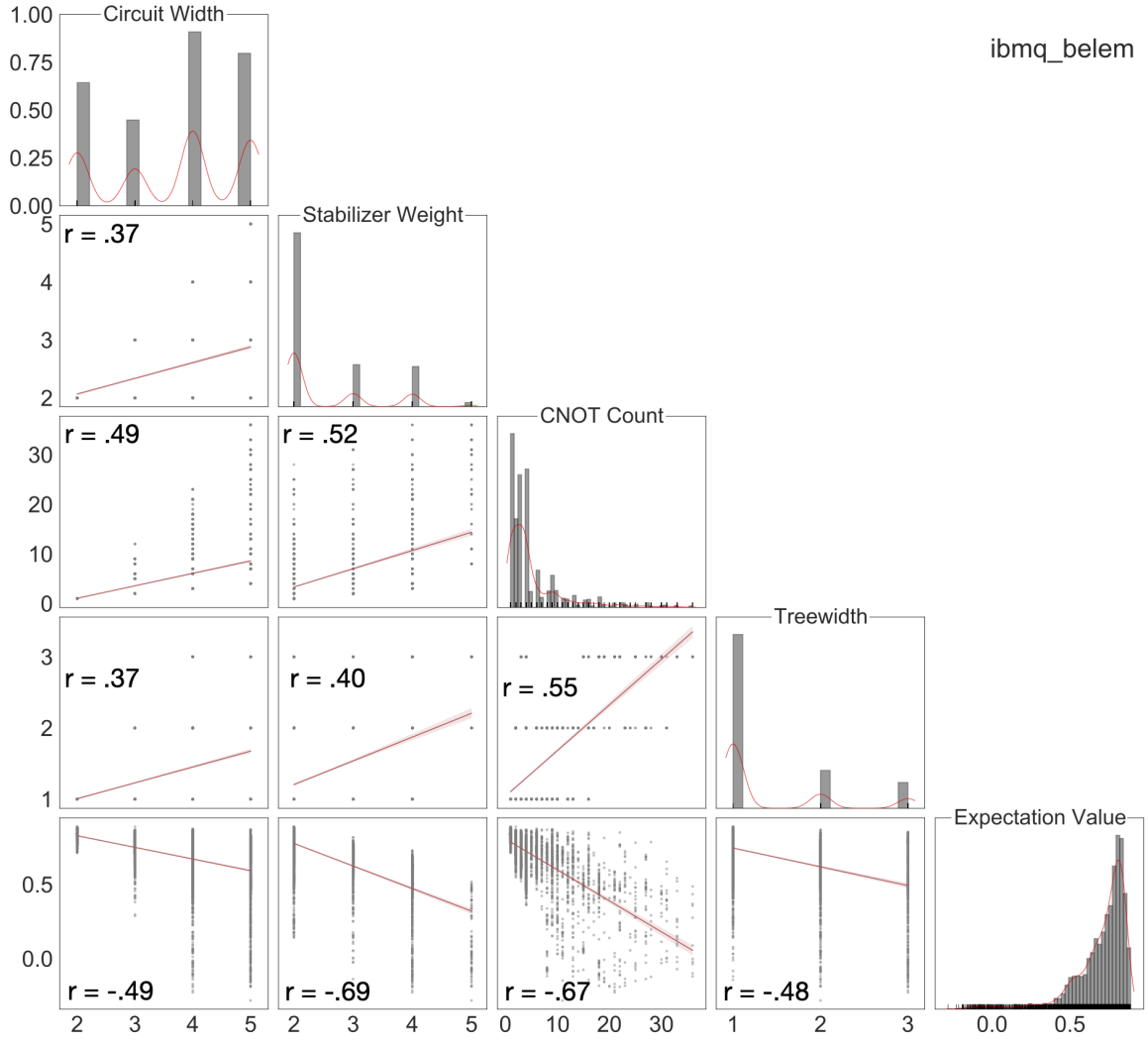
FIG. 17: Correlation Matrix for `ibmq_belem`. Linear regression fit plotted in red with confidence interval and Pearson's r-coefficient computed between features are reported on individual scatter plots. `ibmq_belem` r-coefficient values were computed with 5650 degrees of freedom.