Article

Combining quantum processors with real-time classical communication

https://doi.org/10.1038/s41586-024-08178-2

Received: 21 March 2024

Accepted: 8 October 2024

Published online: 20 November 2024

Open access

Check for updates

Almudena Carrera Vazquez[1], Caroline Tornow[1,2], Diego Ristè[3], Stefan Woerner[1], Maika Takita[4] & Daniel J. Egger[1]✉

Quantum computers process information with the laws of quantum mechanics. Current quantum hardware is noisy, can only store information for a short time and is limited to a few quantum bits, that is, qubits, typically arranged in a planar connectivity[1]. However, many applications of quantum computing require more connectivity than the planar lattice offered by the hardware on more qubits than is available on a single quantum processing unit (QPU). The community hopes to tackle these limitations by connecting QPUs using classical communication, which has not yet been proven experimentally. Here we experimentally realize error-mitigated dynamic circuits and circuit cutting to create quantum states requiring periodic connectivity using up to 142 qubits spanning two QPUs with 127 qubits each connected in real time with a classical link. In a dynamic circuit, quantum gates can be classically controlled by the outcomes of mid-circuit measurements within run-time, that is, within a fraction of the coherence time of the qubits. Our real-time classical link enables us to apply a quantum gate on one QPU conditioned on the outcome of a measurement on another QPU. Furthermore, the error-mitigated control flow enhances qubit connectivity and the instruction set of the hardware thus increasing the versatility of our quantum computers. Our work demonstrates that we can use several quantum processors as one with error-mitigated dynamic circuits enabled by a real-time classical link.

Quantum computers process information encoded in quantum bits with unitary operations. However, quantum computers are noisy and most large-scale architectures arrange the physical qubits in a planar lattice. Nevertheless, current processors with error mitigation can already simulate hardware-native Ising models with 127 qubits and measure observables at a scale where brute-force approaches with classical computers begin to struggle[1]. The usefulness of quantum computers hinges on further scaling and overcoming their limited qubit connectivity. A modular approach is important for scaling current noisy quantum processors[2] and for achieving the large numbers of physical qubits needed for fault tolerance[3]. Trapped ion and neutral atom architectures can achieve modularity by physically transporting the qubits[4,5]. In the near term, modularity in superconducting qubits[6] is achieved by short-range interconnects that link adjacent chips[7,8].

In the medium term, long-range gates operating in the microwave regime may be carried out over long conventional cables[9–11]. This would enable non-planar qubit connectivity suitable for efficient error correction[3]. A long-term alternative is to entangle remote QPUs with an optical link leveraging a microwave to optical transduction[12], which has not yet been demonstrated, to our knowledge. Moreover, dynamic circuits broaden the set of operations of a quantum computer by performing mid-circuit measurements (MCMs) and classically controlling a gate within the coherence time of the qubits. They enhance algorithmic quality[13] and qubit connectivity[14]. As we will show, dynamic circuits also enable modularity by connecting QPUs in real time through a classical link.

We take a complementary approach based on virtual gates to implement long-range interactions in a modular architecture. We connect qubits at arbitrary locations and create the statistics of entanglement through a quasi-probability decomposition (QPD)[15–17]. We compare a Local Operations (LO) only scheme[16] to one augmented by Classical Communication (LOCC)[17]. The LO scheme, demonstrated in a two-qubit setting[18], requires executing multiple quantum circuits with local operations only. By contrast, to implement LOCC, we consume virtual Bell pairs in a teleportation circuit to create two-qubit gates[19,20]. On quantum hardware with sparse and planar connectivity, creating a Bell pair between arbitrary qubits requires a long-range controlled-NOT (CNOT) gate. To avoid these gates, we use a QPD over local operations resulting in cut Bell pairs that the teleportation consumes. LO do not need the classical link and is thus simpler to implement than LOCC. However, as LOCC only requires a single parameterized template circuit, it is more efficient to compile than LO and the cost of its QPD is lower than the cost of the LO scheme.

Our work makes four key contributions. First, we present the quantum circuits and QPD to create multiple cut Bell pairs to realize the virtual gates in ref. 17. Second, we suppress and mitigate the errors arising from the latency of the classical control hardware in dynamic circuits[21] with a combination of dynamical decoupling and zero-noise extrapolation[22]. Third, we leverage these methods to engineer periodic boundary conditions on a 103-node graph state. Fourth, we demonstrate a real-time classical connection between two separate QPUs thereby

[1]IBM Quantum, IBM Research Europe - Zurich, Rüschlikon, Switzerland. [2]Institute for Theoretical Physics, ETH Zurich, Zurich, Switzerland. [3]IBM Quantum, IBM Research Cambridge, Cambridge, MA, USA. [4]IBM Quantum, T. J. Watson Research Center, Yorktown Heights, NY, USA. ✉e-mail: deg@zurich.ibm.com

demonstrating that a system of distributed QPUs can be operated as one through a classical link[23]. Combined with dynamic circuits, this enables us to operate both chips as a single quantum computer, which we exemplify by engineering a periodic graph state that spans both devices on 142 qubits. We discuss a path forward to create long-range gates and provide our conclusion.

Circuit cutting

We run large quantum circuits that may not be directly executable on our hardware because of limitations in qubit count or connectivity by cutting gates. Circuit cutting decomposes a complex circuit into subcircuits that can be individually executed[15–17,24–26]. However, we must run an increased number of circuits, which we call the sampling overhead. The results from these subcircuits are then classically recombined to yield the result of the original circuit (Methods).

As one of the main contributions of our work is implementing virtual gates with LOCC, we show how to create the required cut Bell pairs with local operations. Here, multiple cut Bell pairs are engineered by parameterized quantum circuits, which we call a cut Bell pair factory (Fig. 1b,c). Cutting multiple pairs at the same time requires a lower sampling overhead[17]. As the cut Bell pair factory forms two disjoint quantum circuits, we place each subcircuit close to qubits that have long-range gates. The resulting resource is then consumed in a teleportation circuit. For instance, in Fig. 1b, the cut Bell pairs are consumed to create CNOT gates on the qubit pairs (0, 1) and (2, 3) (see section 'Cut Bell pair factories').

Periodic boundary conditions

We construct a graph state $|G\rangle$ with periodic boundary conditions on ibm_kyiv, an Eagle processor[1], going beyond the limits imposed by its physical connectivity (see section 'Graph states'). Here, G has $|V| = 103$ nodes and requires four long-range edges $E_{\mathrm{lr}} = \{(1, 95), (2, 98), (6, 102), (7, 97)\}$ between the top and bottom qubits of the Eagle processor (Fig. 2a). We measure the node stabilizers S_i at each node $i \in V$ and the edge stabilizers formed by the product S_iS_j across each edge $(i,j) \in E$. From these stabilizers, we build an entanglement witness $\mathcal{W}_{i,j} = (1 - \langle S_i\rangle - \langle S_j\rangle - \langle S_iS_j\rangle)/4$, which is negative if there is bipartite entanglement across the edge $(i,j) \in E$ (ref. 27) (see section 'Entanglement witness'). We focus on bipartite entanglement because this is the resource we wish to recreate with virtual gates. Measuring witnesses of entanglement between more than two parties will measure only the quality of the non-virtual gates and measurements making the impact of the virtual gates less clear.

We prepare $|G\rangle$ using three different methods. The hardware-native edges are always implemented with CNOT gates but the periodic boundary conditions are implemented with (1) SWAP gates, (2) LOCC and (3) LO to connect qubits across the whole lattice. The main difference between LOCC and LO is a feed-forward operation consisting of single-qubit gates conditioned on $2n$ measurement outcomes, where n is the number of cuts. Each of the 2^{2n} cases triggers a unique combination of X and/or Z gates on the appropriate qubits. Acquiring the measurement results, determining the corresponding case and acting based on it is performed in real time by the control hardware, at the cost of a fixed added latency. We mitigate and suppress the errors resulting from this latency with zero-noise extrapolation[22] and staggered dynamical decoupling[21,28] (see section 'Error-mitigated quantum circuit switch instructions').

We benchmark the SWAP, LOCC and LO implementations of $|G\rangle$ with a hardware-native graph state on $G' = (V, E')$ obtained by removing the long-range gates, that is, $E' = E \backslash E_{\mathrm{lr}}$. The circuit preparing $|G'\rangle$ thus requires only 112 CNOT gates arranged in three layers following the heavy-hexagonal topology of the Eagle processor. This circuit will report large errors when measuring the node and edge stabilizers of $|G\rangle$ for nodes on a cut gate because it is designed to implement $|G'\rangle$.

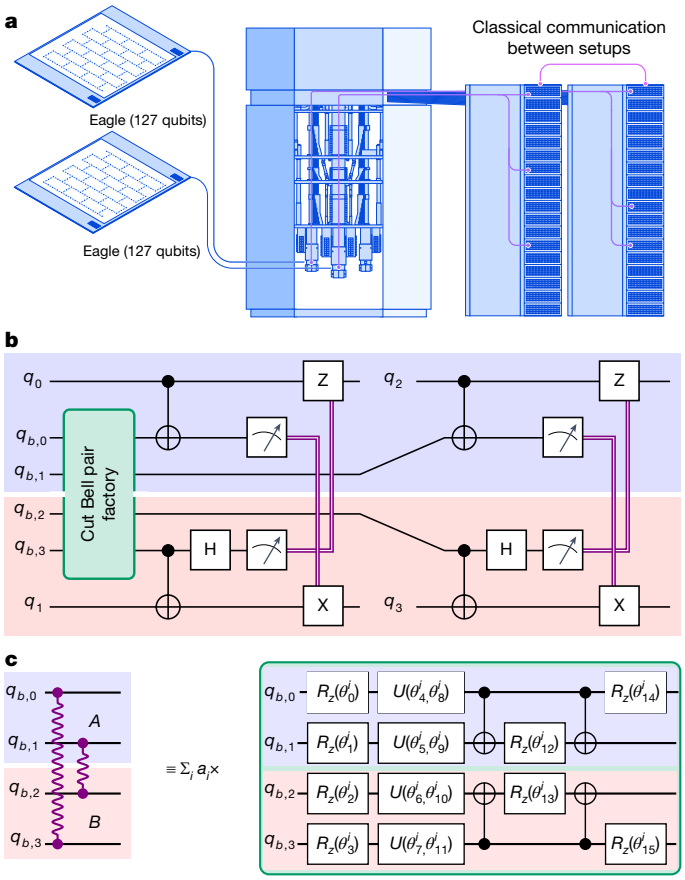


Fig. 1 | Local operations and classical communication. a, Depiction of an IBM Quantum System Two architecture. Here, two 127 qubit Eagle QPUs are connected with a real-time classical link. Each QPU is controlled by its electronics in its rack. We tightly synchronize both racks to operate both QPUs as one. **b**, Template quantum circuit to implement virtual CNOT gates on qubit pairs (q_0, q_1) and (q_2, q_3) with LOCC by consuming cut Bell pairs in a teleportation circuit. The purple double lines correspond to the real-time classical link. **c**, Cut Bell pair factories $C_2(\boldsymbol{\theta}^i)$ for two simultaneously cut Bell pairs. The QPD has a total of 27 different parameter sets $\boldsymbol{\theta}^i$. Here, $U(\theta, \phi) = \sqrt{X}R_z(\theta)\sqrt{X}R_z(\phi)$.

We refer to this hardware-native benchmark as the dropped edge benchmark. The swap-based circuit requires an additional 262 CNOT gates to create the long-range edges E_{lr}, which drastically reduces the value of the measured stabilizers (Fig. 2b–d). By contrast, the LOCC and LO implementation of the edges in E_{lr} does not require SWAP gates. The errors of their node and edge stabilizers for nodes not involved in a cut gate closely follow the dropped edge benchmark (Fig. 2b,c). Conversely, the stabilizers involving a virtual gate have a lower error than the dropped edge benchmark and the swap implementation (Fig. 2c, star markers). As an overall quality metric, we first report the sum of absolute errors on the node stabilizers, that is, $\sum_{i\in V}|S_i - 1|$ (Extended Data Table 1). The large SWAP overhead is responsible for the 44.3 sum absolute error. The 13.1 error on the dropped edge benchmark is dominated by the eight nodes on the four cuts (Fig. 2c, star markers). By contrast, the LO and LOCC errors are affected by MCMs. We attribute the 1.9 additional error of LOCC over LO to the delays and the CNOT gates in the teleportation circuit and cut Bell pairs. In the SWAP-based results, $\mathcal{W}_{i,j}$ does not detect entanglement across 35 of the 116 edges at the 99% confidence level (Fig. 2b,d). For the LO and LOCC implementation, $\mathcal{W}_{i,j}$ witnesses the statistics of bipartite entanglement across all edges in G at the 99% confidence level (Fig. 2e). These metrics show that virtual long-range gates produce stabilizers with smaller errors than their

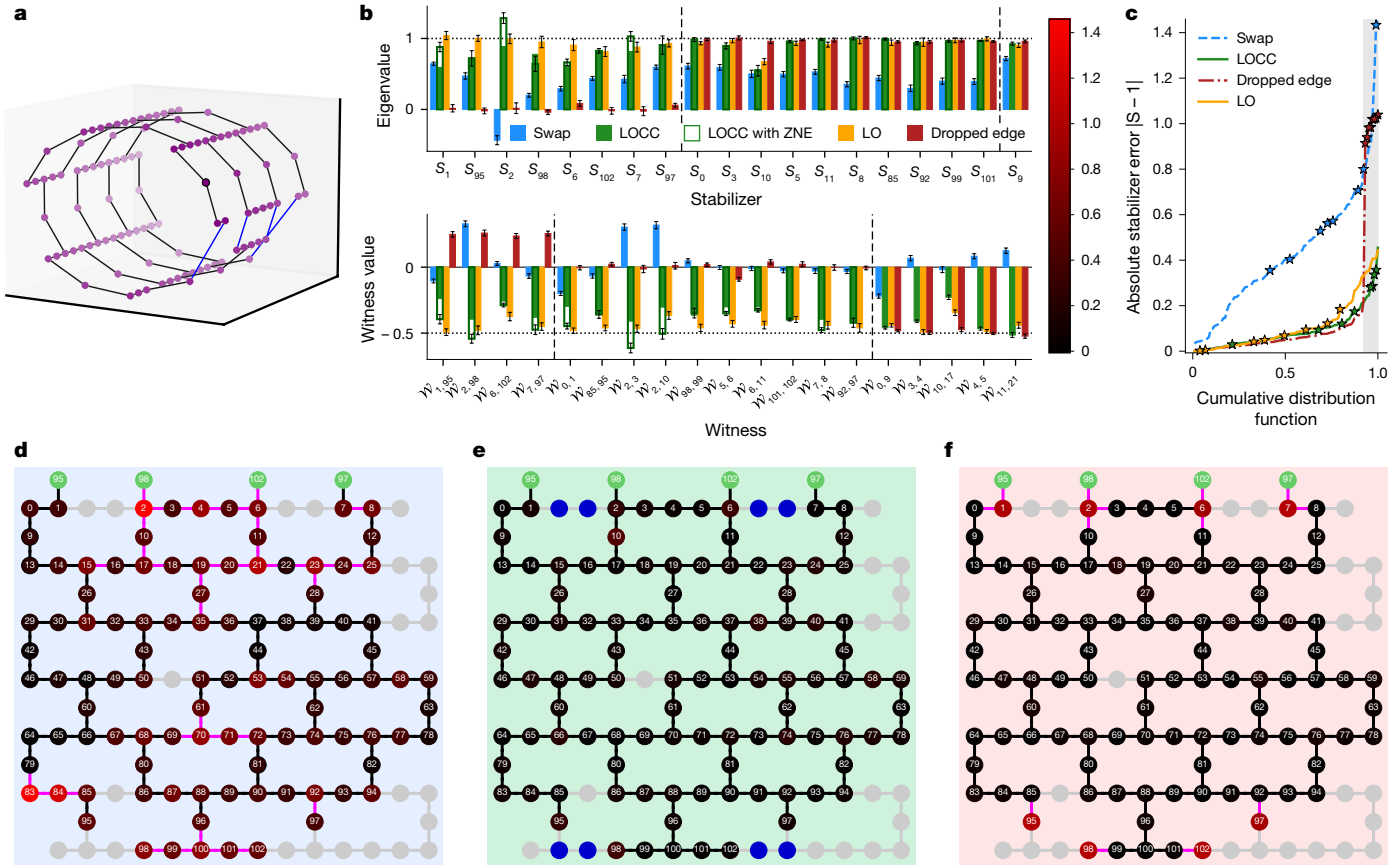


Fig. 2 | Periodic boundary conditions. a, The heavy-hexagonal graph is folded on itself into a tubular form by the edges (1, 95), (2, 98), (6, 102) and (7, 97) highlighted in blue. We cut these edges. **b**, The node stabilizers S_j (top) and witnesses $\mathcal{W}_{i,j}$, (bottom), with 1 standard deviation for the nodes and edges close to the long-range edges. Vertical dashed lines group stabilizers and witnesses by their distance to cut edges. **c**, Cumulative distribution function of the stabilizer errors. The stars indicate node stabilizers S_j that have an edge implemented by a long-range gate. In the dropped edge benchmark (dash-dotted red line), the long-range gates are not implemented and the star-indicated stabilizers thus have unit error. The grey region is the probability mass corresponding to node stabilizers affected by the cuts. **d**–**f**, In the two-dimensional layouts, the green nodes duplicate nodes 95, 98, 102 and 97 to show the cut edges. The blue nodes in **e** are qubit resources to create cut Bell pairs. The colour of node i is the absolute error $|S_i - 1|$ of the measured stabilizer, as indicated by the colour bar. An edge is black if entanglement statistics are detected at a 99% confidence level and violet if not. In **d**, the long-range gates are implemented with SWAP gates. In **e**, the same gates are implemented with LOCC. In **f**, they are not implemented at all.

decomposition into SWAPs. Furthermore, they keep the variance low enough to verify the statistics of entanglement.

Operating two QPUs as one

We now combine two Eagle QPUs with 127 qubits each into a single QPU through a real-time classical connection. Operating the devices as a single, larger processor consists of executing quantum circuits spanning the larger qubit register. Apart from unitary gates and measurements running concurrently on the merged QPU, we use dynamic circuits to perform gates that act on qubits on both devices. This is enabled by a tight synchronization and fast classical communication between physically separate instruments required to collect measurement results and determine the control flow across the whole system[29].

We test this real-time classical connection by engineering a graph state on 134 qubits built from heavy-hexagonal rings that wind through both QPUs (Fig. 3). These rings were chosen by excluding qubits plagued by two-level systems and readout issues to ensure a high-quality graph state. This graph forms a ring in three dimensions and requires four long-range gates that we implement with LO and LOCC. As before, the LOCC protocol thus requires two additional qubits per cut gate for the cut Bell pairs. As in the previous section, we benchmark our results to a graph that does not implement the edges that span both QPUs. As there is no quantum link between the two devices, a benchmark with SWAP gates is impossible. All edges exhibit the statistics of bipartite entanglement when we implement the graph with LO and LOCC at a 99% confidence level. Furthermore, the LO and LOCC stabilizers have the same quality as the dropped edge benchmark for nodes that are not affected by a long-range gate (Fig. 3c). Stabilizers affected by long-range gates have a large reduction in error compared with the dropped edge benchmark. The sum of absolute errors on the node stabilizers $\sum_{i\in V}|S_i - 1|$, is 21.0, 19.2 and 12.6 for the dropped edge benchmark, LOCC and LO, respectively. As before, we attribute the 6.6 additional errors of LOCC over LO to the delays and the CNOT gates in the teleportation circuit and cut Bell pairs. The LOCC results demonstrate how a dynamic quantum circuit in which two subcircuits are connected by a real-time classical link can be executed on two otherwise disjoint QPUs. The LO results could be obtained on a single device with 127 qubits at the cost of an additional factor of 2 in run-time as the subcircuits can be run successively.

Discussion and conclusion

We implement long-range gates with LO and LOCC. With these gates, we engineer periodic boundary conditions on a 103-node planar lattice and connect two Eagle processors in real time to create a graph state on 134 qubits, going beyond the abilities of a single chip. Here, we chose to implement graph states as an application to highlight the scalable

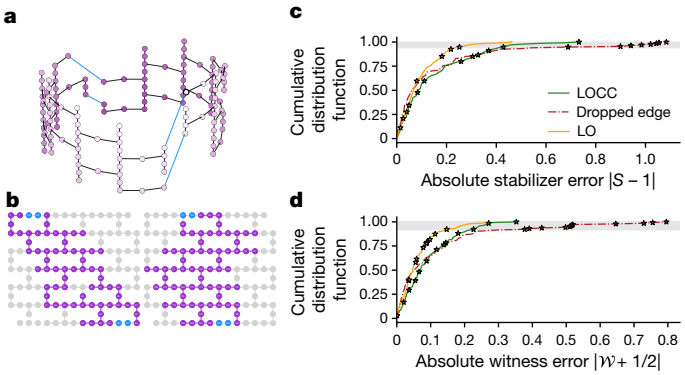


Fig. 3 | Two connected QPUs with LOCC. a, Graph state with periodic boundaries shown in three dimensions. The blue edges are the cut edges. **b**, Coupling map of two Eagle QPUs operated as a single device with 254 qubits. The purple nodes are the qubits forming the graph state in **a** and the blue nodes are used for cut Bell pairs. **c**,**d**, Absolute error on the stabilizers (**c**) and edge witnesses (**d**) implemented with LOCC (solid green) and LO (solid orange) and on a dropped edge benchmark graph (dotted-dashed red) for the graph state in **a**. In **c** and **d**, the stars show stabilizers and edge witnesses that are affected by the cuts. In **c** and **d**, the grey region is the probability mass corresponding to node stabilizers and edge witnesses, respectively, affected by the cut. In **c** and **d**, we observe that the LO implementation outperforms the dropped edge benchmark, which we attribute to better device conditions as these data were taken on a different day from the benchmark and LOCC data.

properties of dynamic circuits. Our cut Bell pair factories enable the LOCC scheme presented in ref. 17. Both the LO and LOCC protocols deliver high-quality results that closely match a hardware-native benchmark. Circuit cutting increases the variance of measured observables. We can keep the variance under control in both the LO and LOCC schemes as indicated by the statistical tests on the witnesses. An in-depth discussion of the measured variance is found in the Supplementary Information.

The variance increase from the QPD is why research now focuses on reducing the sampling overhead. It was recently shown that cutting multiple two-qubit gates in parallel results in optimal LO QPDs with the same sampling overhead as LOCC but requires an additional ancilla qubit and possibly reset[30,31]. In LOCC, the QPD is required only to cut the Bell pairs. This costly QPD could be removed, that is, no shot overhead, by distributing entanglement across multiple chips[32,33]. In the near to medium term, this could be done by operating gates in the microwave regime over conventional cables[10,34,35] or, in the long term, with an optical-to-microwave transduction[36–38]. Entanglement distribution is typically noisy and may result in non-maximally entangled states. However, gate teleportation requires a maximally entangled resource. Nevertheless, non-maximally entangled states could lower the sampling cost of the QPD[39] and multiple copies of non-maximally entangled states could be distilled into a pure state for teleportation[40] either during the execution of a quantum circuit or possibly during the delays between consecutive shots, which may be as large as 250 μs for resets[41]. Combined with these settings, our error-mitigated and suppressed dynamic circuits would enable a modular quantum computing architecture without the sampling overhead of circuit cutting.

In an application setting, circuit cutting could benefit Hamiltonian simulation[42]. Here, the cost of circuit cutting is exponential in the strength of the cut bonds times the evolution time. This cost may thus be reasonable for weak bonds and/or short evolution times. Furthermore, the LO scheme presented in ref. 42 requires ancilla qubits in a Hadamard test, which would require a reset through a dynamic circuit if the same bond is cut multiple times in a Trotterized time evolution.

Circuit cutting can be applied to both wires and gates. The resulting quantum circuits have a similar structure making our approach applicable to both cases. Our real-time classical link implements long-range gates and classically couples disjoint quantum processors. The cut Bell pairs that we present have values beyond our work. For example, these pairs are directly usable to cut circuits in measurement-based quantum computing, which relies on dynamic circuits[14]. This could also be accomplished with LO; the result would be an execution setting identical to ours with dynamic circuits. Furthermore, the combination of staggered dynamical decoupling with zero-noise extrapolation mitigates the lengthy delays of the feed-forward operations, which enables a high-quality implementation of dynamic circuits. Our work sheds light on the noise sources, such as *ZZ* cross-talk occurring during the latency, that a transpiler for distributed superconducting quantum computers must consider[43]. In summary, we demonstrate that we can use several quantum processors as one with error-mitigated dynamic circuits enabled by a real-time classical link.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-024-08178-2.

1. Kim, Y. et al. Evidence for the utility of quantum computing before fault tolerance. *Nature* **618**, 500–505 (2023).
2. Bravyi, S., Dial, O., Gambetta, J. M., Gil, D. & Nazario, Z. The future of quantum computing with superconducting qubits. *J. Appl. Phys.* **132**, 160902 (2022).
3. Bravyi, S. et al. High-threshold and low-overhead fault-tolerant quantum memory. *Nature* **627**, 778–782 (2024).
4. Akhtar, M. et al. A high-fidelity quantum matter-link between ion-trap microchip modules. *Nat. Commun.* **14**, 531 (2023).
5. Bluvstein, D. et al. A quantum processor based on coherent transport of entangled atom arrays. *Nature* **604**, 451–456 (2022).
6. Krantz, P. et al. A quantum engineer's guide to superconducting qubits. *Appl. Phys. Rev.* **6**, 021318 (2019).
7. Conner, C. R. et al. Superconducting qubits in a flip-chip architecture. *Appl. Phys. Lett.* **118**, 232602 (2021).
8. Gold, A. et al. Entanglement across separate silicon dies in a modular superconducting qubit device. *npj Quantum Inf.* **7**, 142 (2021).
9. Zhong, Y. et al. Violating Bell's inequality with remotely connected superconducting qubits. *Nat. Phys.* **15**, 741–744 (2019).
10. Zhong, Y. et al. Deterministic multi-qubit entanglement in a quantum network. *Nature* **590**, 571–575 (2021).
11. Malekakhlagh, M. et al. Enhanced quantum state transfer and Bell-state generation over long-range multimode interconnects via superadiabatic transitionless driving. *Phys. Rev. Appl.* **22**, 024006 (2024).
12. Ang, J. et al. ARQUIN: architectures for multinode superconducting quantum computers. *ACM Trans. Quantum Comput.* **5**, 19 (2024).
13. Córcoles, A. D. et al. Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits. *Phys. Rev. Lett.* **127**, 100501 (2021).
14. Bäumer, E. et al. Efficient long-range entanglement using dynamic circuits. *PRX Quantum* **5**, 030339 (2024).
15. Hofmann, H. F. How to simulate a universal quantum computer using negative probabilities. *J. Phys. A Math. Theor.* **42**, 275304 (2009).
16. Mitarai, K. & Fujii, K. Constructing a virtual two-qubit gate by sampling single-qubit operations. *New J. Phys.* **23**, 023021 (2021).
17. Piveteau, C. & Sutter, D. Circuit knitting with classical communication. *IEEE Trans. Inf. Theor.* **1**, 2734–2745 (2023).
18. Singh, A. P. et al. Experimental demonstration of a high-fidelity virtual two-qubit gate. *Phys. Rev. Res.* **6**, 013235 (2024).
19. Gottesman, D. & Chuang, I. L. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature* **402**, 390–393 (1999).
20. Wan, Y. et al. Quantum gate teleportation between separated qubits in a trapped-ion processor. *Science* **364**, 875 (2019).
21. Viola, L., Knill, E. & Lloyd, S. Dynamical decoupling of open quantum systems. *Phys. Rev. Lett.* **82**, 2417–2421 (1999).
22. Temme, K., Bravyi, S. & Gambetta, J. M. Error mitigation for short-depth quantum circuits. *Phys. Rev. Lett.* **119**, 180509 (2017).
23. Reilly, D. J. Challenges in scaling-up the control interface of a quantum computer. Preprint at arxiv.org/abs/1912.05114 (2019).
24. Peng, T., Harrow, A. W., Ozols, M. & Wu, X. Simulating large quantum circuits on a small quantum computer. *Phys. Rev. Lett.* **125**, 150504 (2020).
25. Brenner, L., Piveteau, C. & Sutter, D. Optimal wire cutting with classical communication. Preprint at arxiv.org/abs/2302.03366 (2023).
26. Pednault, E. An alternative approach to optimal wire cutting without ancilla qubits. Preprint at arxiv.org/abs/2303.08287 (2023).

27. Zander, R. & Becker, C. K.-U. Benchmarking multipartite entanglement generation with graph states. Preprint at arxiv.org/abs/2402.00766 (2024).
28. Mundada, P. S. et al. Experimental benchmarking of an automated deterministic error-suppression workflow for quantum algorithms. *Phys. Rev. Appl.* **20**, 024034 (2023).
29. Gupta, R. S. et al. Encoding a magic state with beyond break-even fidelity. *Nature* **625**, 259–263 (2024).
30. Ufrecht, C. et al. Optimal joint cutting of two-qubit rotation gates. *Phys. Rev. A* **109**, 052440 (2024).
31. Schmitt, L., Piveteau, C. & Sutter, D. Cutting circuits with multiple two-qubit unitaries. Preprint at arxiv.org/abs/2312.11638 (2023).
32. Cirac, J. I., Zoller, P., Kimble, H. J. & Mabuchi, H. Quantum state transfer and entanglement distribution among distant nodes in a quantum network. *Phys. Rev. Lett.* **78**, 3221–3224 (1997).
33. Duan, L.-M., Lukin, M. D., Cirac, J. I. & Zoller, P. Long-distance quantum communication with atomic ensembles and linear optics. *Nature* **414**, 413–418 (2001).
34. Magnard, P. et al. Microwave quantum link between superconducting circuits housed in spatially separated cryogenic systems. *Phys. Rev. Lett.* **125**, 260502 (2020).
35. Niu, J. et al. Low-loss interconnects for modular superconducting quantum processors. *Nat. Electron.* **6**, 235–241 (2023).
36. Orcutt, J. et al. Engineering electro-optics in SiGe/Si waveguides for quantum transduction. *Quantum Sci. Technol.* **5**, 034006 (2020).
37. Lauk, N. et al. Perspectives on quantum transduction. *Quantum Sci. Technol.* **5**, 020501 (2020).
38. Krastanov, S. et al. Optically heralded entanglement of superconducting systems in quantum networks. *Phys. Rev. Lett.* **127**, 040503 (2021).
39. Bechtold, M., Barzen, J., Leymann, F. & Mandl, A. Circuit cutting with non-maximally entangled states. Preprint at arxiv.org/abs/2306.12084 (2023).
40. Bennett, C. H. et al. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.* **76**, 722–725 (1996).
41. Tornow, C., Kanazawa, N., Shanks, W. E. & Egger, D. J. Minimum quantum run-time characterization and calibration via restless measurements with dynamic repetition rates. *Phys. Rev. Appl.* **17**, 064061 (2022).
42. Harrow, A. W. & Lowe, A. Optimal quantum circuit cuts with application to clustered Hamiltonian simulation. Preprint at arxiv.org/abs/2403.01018 (2024).
43. Caleffi, M. et al. Distributed quantum computing: a survey. Preprint at arxiv.org/abs/2212.10609 (2022).

Methods

Circuit cutting

The gates in a quantum circuit are quantum channels acting on density matrices ρ. A single quantum channel $\mathcal{E}(\rho)$ is cut by expressing it as a sum over I quantum channels $\mathcal{E}_i(\rho)$ resulting in the QPD

$$\mathcal{E}(\rho) = \sum_{i=0}^{I-1} a_i \mathcal{E}_i(\rho). \tag{1}$$

The channels $\mathcal{E}_i(\rho)$ are easier to implement than $\mathcal{E}(\rho)$ and are built from LO[16] or LOCC[17] (Fig. 1). As some of the coefficients a_i are negative, we introduce $\gamma = \sum_i |a_i|$ and $P_i = |a_i|/\gamma$ to recover a valid probability distribution with probabilities P_i over the channels $\mathcal{E}_i$. Here, γ can be seen as the amount by which the QPD deviates from a true probability distribution and is thus a cost to pay to implement the QPD. Without a QPD an observable is estimated by $\langle O\rangle = \mathrm{Tr}\{O\mathcal{E}(\rho)\}$. However, when using this QPD, we build an unbiased Monte Carlo estimator of O as

$$\langle O\rangle_{\mathrm{QPD}} = \gamma \sum_{i=0}^{I-1} P_i \mathrm{sign}(a_i)\ \mathrm{Tr}\{O\mathcal{E}_i(\rho)\}. \tag{2}$$

The variance of the QPD estimator $\langle O\rangle_{\mathrm{QPD}}$ is a factor of γ^2 larger than the variance of the non-cut estimator $\langle O\rangle$ (ref. 44). When cutting $n > 1$ identical channels, we can build an estimator by taking the product of the QPDs for each individual channel, resulting in a γ^{2n} rescaling factor[22,45]. This exponential increase in variance is compensated by a corresponding increase in the number of measured shots. Therefore, γ^{2n} is called the sampling overhead and indicates that circuit cutting must be used sparingly. Details of the LO and LOCC quantum channels $\mathcal{E}_i$ and their coefficients a_i are provided in sections 'Virtual gates implemented with LO' and 'Virtual gates implemented with LOCC', respectively.

Virtual gates implemented with LO

Here, we discuss how to implement virtual CZ gates with LO[16,18]. We follow ref. 16 and, therefore, decompose each cut CZ gate into local operations and a sum over six different circuits defined by

$$\begin{aligned} 2\mathrm{CZ} = &\sum_{\alpha\in\{\pm1\}} R_z\left(\alpha\frac{\pi}{2}\right)\otimes R_z\left(\alpha\frac{\pi}{2}\right) \\ &- \sum_{\alpha_1,\alpha_2\in\{\pm1\}} \alpha_1\alpha_2 R_z\left(-\frac{\alpha_1+1}{2}\pi\right)\otimes\left(\frac{I+\alpha_2 Z}{2}\right) \\ &- \sum_{\alpha_1,\alpha_2\in\{\pm1\}} \alpha_1\alpha_2\left(\frac{I+\alpha_1 Z}{2}\right)\otimes R_z\left(-\frac{\alpha_2+1}{2}\pi\right), \end{aligned} \tag{3}$$

where $R_z(\theta) = \exp\left(-\mathrm{i}\frac{\theta}{2}Z\right)$ are virtual Z rotations[46]. The factor 2 in front of CZ is for readability. Each of the possible six circuits is thus weighted by a 1/6 probability (Extended Data Fig. 1). The operations $(I+Z)/2$ and $(I-Z)/2$ correspond to the projectors $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$, respectively. They are implemented by MCMs and classical post-processing. More specifically, when computing the expectation value of an observable $\langle O\rangle = \sum_i a_i\langle O\rangle_i$ with the LO QPD, we multiply the expectation values $\langle O\rangle_i$ by 1 and −1 when the outcome of an MCM is 0 and 1, respectively.

In the experiments that implement graph states with LO in the main text, we implement the CZ gate with six circuits built from R_z gates and MCMs[16]. Cutting four CZ gates with LO thus requires $I = 6^4 = 1{,}296$ circuits. However, as the node and edge stabilizers of the graph states are at most in the light cone[47] of one virtual gate, we instead implement two QPDs in parallel, which requires $I = 6^2 = 36$ LO circuits per expectation value. In general, sampling from a QPD results in an overhead of $(\sum_{i=0}^{I-1}|a_i|)^2$, where I is the number of circuits in the QPD and the a_i are the QPD coefficients[44]. However, as the LO QPDs in our experiments have only 36 circuits, we fully enumerate the QPDs by executing all 36 circuits. The sampling cost of full enumeration is $I(\sum_{i=0}^{I-1}|a_i|^2)$. Furthermore, as $|a_i| = 1/2\ \forall\ i = 0, \ldots, I-1$, sampling from the QPD and fully enumerating it both have the same shot overhead.

The decomposition in equation (3) with $\gamma^2 = 9$ is optimal with respect to the sampling overhead for a single gate[17]. Recently, refs. 30,31 found a new protocol that achieves the same γ overhead as LOCC when cutting multiple gates in parallel. The proofs in refs. 30,31 are theoretical demonstrating the existence of a decomposition.

Virtual gates implemented with LOCC

We now discuss the implementation of the dynamic circuits that enable the virtual gates with LOCC. We first present an error suppression and mitigation of dynamic circuits with dynamical decoupling (DD) and zero-noise extrapolation (ZNE). Second, we discuss the methodology to create the cut Bell pairs and present the circuits to implement one, two and three cut Bell pairs. Finally, we propose a simple benchmark experiment to assess the quality of a virtual gate.

Error-mitigated quantum circuit switch instructions. All quantum circuits presented in this work are written in Qiskit. The feed-forward operations of the LOCC circuits are executed with a quantum circuit switch instruction, hereafter referred to as a switch. A switch defines a set of cases in which the quantum circuit can branch depending on the outcome of a corresponding set of measurements. This branching occurs in real time for each experimental shot, with the measurement outcomes being collected by a central processor, which in turn broadcasts the selected case (here corresponding to a combination of X and Z gates) to all control instruments.

As quantum computing scales, the control electronics become tailored to its QPU and are no longer built from off-the-shelf components. Recent IBM devices have a single QPU with a rack of dedicated and tailored control electronics, as shown in refs. 29,48. The realization of the feed-forward we present builds upon the work in ref. 29 and advances its scalability in two main ways. First, our development enables the synchronization and inter-communication between separate experimental setups. Not only are the control instruments for the two sub-QPUs located in different racks, but they are also configurable in software to operate on them independently for the LO experiments and recombined for LOCC. This architecture is extensible to multiple racks and QPUs. It overcomes several of the challenges in operating a distributed control system as pointed out in ref. 23. Second, the duration of the conditional operation is independent of the measurement results, of which qubits are measured, and which qubits are subject to the conditional operations (apart from minor differences due to cable lengths). This enables the scheduling and execution of programs equally across the combined QPU as if it were a single one.

The feed-forward process results in a latency of the order of 0.5 µs (independent of the selected case) during which no gates can be applied (Extended Data Fig. 2a, red area). Free evolution during this period (τ), often dominated by static ZZ cross-talk in the Hamiltonian, typically with a strength ranging from about 10^3 Hz to 10^4 Hz, substantially deteriorates results. To cancel this unwanted interaction and any other constant or slowly fluctuating IZ or ZI terms, we precede the conditional gates with a staggered DD X–X sequence, adding 3τ to the switch duration (Extended Data Fig. 2a). The value of τ is determined by the longest latency path from one QPU to the other and is fine-tuned by maximizing the signal on such a DD sequence. Furthermore, we mitigate the effect of the overall delay on the observables of interest with ZNE[22]. To do this, we first stretch the switch duration by a factor $c = (\tau+\delta)/\tau$, where δ is a variable delay added before each X gate in the DD sequence (Extended Data Fig. 2a). Second, we extrapolate the stabilizer values to the zero-delay limit $c = 0$ with a linear fit. In many cases, an exponential fit can be justified[1]; however, we observe in our benchmark experiments that a linear fit is appropriate (Extended Data Fig. 2). Without DD, we observe strong oscillations in the measured

stabilizers that prevent an accurate ZNE (see the *XZ* stabilizer in Extended Data Fig. 2c). As seen in the main text, this error suppression and mitigation reduce the error on the stabilizers affected by virtual gates.

The error suppression and mitigation that we implement for the switch also apply to other control flow statements. The switch is not the only instruction capable of representing control flow. For instance, OpenQASM3[49] supports if/else statements. Our scheme is done by (1) adding DD sequences to the latency (possibly by adding delays if the control electronics cannot emit pulses during the latency); (2) stretching the delay; and (3) extrapolating to the zero-delay limit.

Cut Bell pair factories. Here, we discuss the quantum circuits to prepare the cut Bell pairs needed to realize virtual gates with LOCC. To create a factory for k cut Bell pairs, we must find a linear combination of circuits with two disjoint partitions with k qubits each to reproduce the statistics of Bell pairs. We create the state ρ_k of the Bell pairs following ref. 50 such that $\rho_k = (1+t_k)\rho_k^+ - t_k\rho_k^-$, where $t_k = 2^k - 1$. Here, $\rho_k^\pm$ are mixed states separable with respect to the partitions A and B. Note that ρ_k entangles the qubit partitions A and B, shown in Fig. 1c, but $\rho_k^\pm$ do not. The total cost of this QPD with two states is determined by $\gamma_k = 2t_k + 1$. Next, we realize $\rho_k^\pm$ from a probabilistic mixture of pure states $\rho_{k,i}^\pm$, that is, valid probability distributions. The state ρ_k^- is easily implemented by a uniform mixture of all basis states that correspond to a 0 entry on the diagonal of the density matrix ρ_k. The basis states themselves do not appear in ρ_k. We thus implement ρ_k^- as a diagonal density matrix of $n_k^- = 4^k - 2^k$ basis states. The state ρ_k^+ is harder to engineer. It requires a probabilistic mixture of intricate states with entanglement within each partition A and B but not between them. To engineer ρ_k^+, we thus build a parametric quantum circuit $C_k(\boldsymbol{\theta}^i)$ with parameters $\boldsymbol{\theta}^i$ in which no two-qubit gate connects qubits between A and B. Following ref. 50, we need $n_k^+ = 2^{2^k} - 1$ pure states to realize ρ_k^+. The exact form of ρ_k^+, omitted here for brevity, is given in Appendix B of ref. 50. Therefore, the total number of parameter sets $I = n_k^+ + n_k^-$ required to implement one, two and three cut Bell pairs is 5, 27 and 311, respectively. Finally, the coefficients $a_{i,k}$ of all the circuits in the QPD in equation (1) that implement $\rho_k^\pm$ are

$$a_{i,k} = \frac{1+t_k}{n_k^+}, \quad \text{for } i \in \{0, \ldots, n_k^+ - 1\}, \text{ and} \tag{4}$$

$$a_{i,k} = -\frac{t_k}{n_k^-}, \quad \text{for } i \in \{n_k^+, \ldots, n_k^+ + n_k^- - 1\}. \tag{5}$$

For $k = 2$, the resulting weights, $|a_{i,k}|/\gamma_k$ are approximately all equal. There is thus no practical difference between sampling and enumerating the $k = 2$ QPD when executing it on hardware. More precisely, for the factories with two cut Bell pairs that we run on hardware, the cost of sampling the QPD is $(\sum_{i=0}^{I-1} |a_{i,2}|)^2 = \gamma_2^2(1 + 1.6 \times 10^{-7})$ and the cost of fully enumerating the QPD is $I(\sum_{i=0}^{I-1} |a_{i,2}|^2) = \gamma_2^2(1 + 1.0 \times 10^{-3})$, where $\gamma_2 = 7$.

We construct all pure states $\rho_{k,i}^\pm$ from the same template variational quantum circuit $C_k(\boldsymbol{\theta}^i)$ with parameters $\boldsymbol{\theta}^i$, where the index $i = 0, \ldots, I-1$ runs over the I elements of the probabilistic mixtures defining $\rho_k^\pm$. The parameters $\boldsymbol{\theta}^i$ in the template circuits $C_k(\boldsymbol{\theta}^i)$ are optimized by the SLSQP classical optimizer[51] by minimizing the L_2-norm with respect to the I pure target states needed to represent ρ_k^+, where the norm is evaluated with a classical state vector simulation. After testing various approaches, we find that those provided in Fig. 1c and Extended Data Fig. 3 enable us to achieve an error, based on the L_2 norm, of less than 10^{-8} for each state while having minimal hardware requirements. To enable rapid execution of the QPD with parametric updates, all the parameters are the angles of virtual Z rotations[46] (Fig. 1c). As ρ_k^- is built from basis states, we analytically derive the parameters. Therefore, we could also significantly simplify the ansatz $C_k(\boldsymbol{\theta}^i)$, for example, by cancelling CNOT gates. However, we keep the same template for compilation and execution efficiency. On first inspection, the parameters entering ρ_k^+ do not have any usable structure. We thus leave it up to future research to further investigate whether these parameters have any structure that could be leveraged to simplify the cut Bell pair factories.

A single-cut Bell pair is engineered by applying the gates $U(\theta_0, \theta_1)$ and $U(\theta_2, \theta_3)$ on qubits 0 and 1. Here, and in the figures, the gate $U(\theta, \phi)$ corresponds to $\sqrt{X}R_z(\theta)\sqrt{X}R_z(\phi)$. The QPD of a single-cut Bell pair requires five sets of parameters given by {[π/2, 0, π/2, 0], [π/2, −2π/3, π/2, 2π/3], [π/2, 2π/3, π/2, −2π/3], [π, 0, 0, 0], [0, 0, π, 0]} which could also be derived analytically. The circuits to simultaneously create two and three cut Bell pairs are shown in Fig. 1c and Extended Data Fig. 3, respectively. The circuits and the values of the parameters as obtained by the optimizer are available on GitHub (www.github.com/eggerdj/cut_graph_state_data).

In the experiments that implement graph states with LOCC in the main text, we construct two QPDs in parallel with $I = 27$ circuits, each QPD implementing two long-range CZ gates. This execution is similar to the LO execution in which we also execute two QPDs in parallel.

Benchmarking qubits for LOCC. The quality of a CNOT gate implemented with dynamic circuits depends on hardware properties. For example, qubit relaxation, dephasing and static ZZ cross-talk all negatively affect the qubits during the idle time of the switch. Furthermore, measurement quality also affects virtual gates implemented with LOCC. Errors on MCMs are harder to correct than errors on final measurements as they propagate to the rest of the circuit through the conditional gates[52]. For instance, assignment errors during readout result in an incorrect application of a single-qubit X or Z gate. Given the variability in these qubit properties, care must be taken in selecting those to act as cut Bell pairs. To determine which qubits will perform well as cut Bell pairs, we develop a fast characterization experiment on four qubits that does not require a QPD or error mitigation. This experiment creates a graph state between qubits 0 and 3 by consuming an uncut Bell pair created on qubits 1 and 2 with a Hadamard and a CNOT gate. We measure the stabilizers ZX and XZ which require two different measurement bases. The resulting circuit, shown in Extended Data Fig. 4a, is structurally equivalent to half of the circuit that consumes two cut Bell pairs, for example, Fig. 1c. We execute this experiment on all qubit chains of length four on the devices that we use and report the mean squared error (MSE), that is, $[(\langle ZX\rangle - 1)^2 + (\langle XZ\rangle - 1)^2]/2$ as a quality metric. The lower the MSE is the better the set of qubits act as cut Bell pairs. With this experiment we benchmark, ibm_kyiv (the device used to create the graph state with 103 nodes), and ibm_pinguino-1a and ibm_pinguino-1b (the two Eagle QPUs combined into a single device, named ibm_pinguino-2a, used to create the graph state with 134 nodes). We observe more than an order of magnitude variation in MSE across each device (Extended Data Fig. 4b).

The qubits we chose to act as cut Bell pairs are a tradeoff between the graph we want to engineer and the quality of the MSE benchmark. For example, the graphs with periodic boundary conditions presented in the main text were designed first based on the desired shape of $|G\rangle$ and second based on the MSE of the Bell pair quality test.

Graph states

A graph state $|G\rangle$ is created from a graph $G = (V, E)$ with nodes V and edges E by applying an initial Hadamard gate to each qubit, corresponding to a node in V, and then CZ gates to each pair of qubits $(i, j) \in E$ (refs. 53,54). The resulting state $|G\rangle$ has $|V|$ first-order stabilizers, one for each node $i \in V$, defined by $S_i = X_i \prod_{k \in N(i)} Z_k$. Here, $N(i)$ is the neighbourhood of node i defined by E. These stabilizers satisfy $S_i|G\rangle = |G\rangle$. By construction, any product of stabilizers is also a stabilizer. If an edge $(i, j) \in E$ is not implemented by a CZ gate, the corresponding stabilizers

drop to zero, that is, $\langle S_i\rangle = \langle S_j\rangle = 0$. This effect can be seen in the dropped edge benchmark, see, for example, Fig. 2b.

Entanglement witness

We now define a success criterion for the implementation of a graph state with entanglement witnesses[55]. A witness $\mathcal{W}$ is designed to detect a certain form of entanglement. As we cut edges in the graph state, we focus on witnesses $\mathcal{W}_{i,j}$ over two nodes i and j connected by an edge in E. An edge (i,j) of our graph state $|G\rangle$ presents entanglement if the expectation value $\langle\mathcal{W}_{i,j}\rangle < 0$. The witness does not detect entanglement if $\langle\mathcal{W}_{i,j}\rangle \geq 0$. The first-order stabilizers of nodes i and j with $(i,j) \in E$ are

$$S_i = Z_jX_i\prod_{k\in N(i)\setminus j} Z_k \text{ and } S_j = X_jZ_i\prod_{k\in N(j)\setminus i} Z_k. \tag{6}$$

Here, $N(i)$ is the neighbourhood of node i, which includes j because $(i,j) \in E$. Thus, $N(i)\setminus j$ is the neighbourhood of node i excluding j. Following refs. 55,56, we build an entanglement witness for edge $(i,j) \in E$ as

$$\mathcal{W}_{i,j} = \frac{1}{4}\mathbb{I} - \frac{1}{4}(\langle S_i\rangle + \langle S_j\rangle + \langle S_iS_j\rangle). \tag{7}$$

This witness is zero or positive if the states are separable. Alternatively, as in ref. 27, a witness for bi-separability is also given by

$$\mathcal{W}'_{i,j} = \mathbb{I} - \langle S_i\rangle - \langle S_j\rangle. \tag{8}$$

Here, we consider both witnesses. The data in the main text are presented for $\mathcal{W}_{i,j}$. As discussed in ref. 56, $\mathcal{W}_{i,j}$ is more robust to noise than $\mathcal{W}'_{i,j}$. However, $\mathcal{W}_{i,j}$ requires more experimental effort to measure than $\mathcal{W}'_{i,j}$ because of the stabilizer S_iS_j.

For completeness, we now show how a witness can detect entanglement by focusing on $\mathcal{W}_{i,j}$. A separable state satisfies $\langle P_1 \ldots P_n\rangle = \prod_i \langle P_i\rangle$, where P_i are single-qubit Pauli operators. Therefore, we can show, using the Cauchy–Schwarz inequality, that $\langle S_i\rangle + \langle S_j\rangle + \langle S_iS_j\rangle \leq 1$ and that $\mathcal{W}_{i,j} \geq 0$ for separable states.

$$\langle S_i\rangle + \langle S_j\rangle + \langle S_iS_j\rangle = \langle Z_j\rangle\langle X_i\rangle\prod_{k\in N(i)\setminus j}\langle Z_k\rangle \tag{9}$$

$$+\langle X_j\rangle\langle Z_i\rangle\prod_{k\in N(j)\setminus i}\langle Z_k\rangle + \langle Y_i\rangle\langle Y_j\rangle\prod_{k\in M(i,j)}\langle Z_k\rangle \tag{10}$$

$$\leq |\langle Z_j\rangle||\langle X_i\rangle| + |\langle X_j\rangle||\langle Z_i\rangle| + |\langle Y_j\rangle||\langle Y_i\rangle| \tag{11}$$

$$\leq \sqrt{\langle X_i\rangle^2 + \langle Y_i\rangle^2 + \langle Z_i\rangle^2}\sqrt{\langle X_j\rangle^2 + \langle Y_j\rangle^2 + \langle Z_j\rangle^2} \tag{12}$$

$$\leq 1. \tag{13}$$

The step from equation (10) to equation (11) relies on $\prod_i a_i \leq \prod_i |a_i|$ and that $\prod_k |\langle Z_k\rangle| \leq 1$, where the product runs over nodes that do not contain i or j. The step from equation (11) to equation (12) is based on the Cauchy–Schwarz inequality. The final step relies on the fact that $\langle X_i\rangle^2 + \langle Y_i\rangle^2 + \langle Z_i\rangle^2 \leq 1$ with pure states equal to one. Therefore, the witness $\mathcal{W}_{i,j}$ will be negative if the state is not separable.

In the graph states presented in the main text, we execute a statistical test at a 99% confidence level to detect entanglement. As discussed in the Supplementary Information and shown in Fig. 2b, some witnesses may go below −1/2 because of readout error mitigation, the QPD and Switch ZNE. We, therefore, consider an edge to have the statistics of entanglement if the deviation from −1/2 is not statistically greater than ±1/2. Based on a one-tailed test, we consider that edge (i,j) is bi-partite entangled if

$$-\frac{1}{2} + \left|\langle\mathcal{W}_{i,j}\rangle + \frac{1}{2}\right| + z_{99\%}\sigma_{\mathcal{W},i,j} < 0. \tag{14}$$

Similarly, we form a success criterion based on $\mathcal{W}'_{i,j}$ as

$$-1 + |\langle\mathcal{W}'_{i,j}\rangle + 1| + z_{99\%}\sigma_{\mathcal{W}',i,j} < 0. \tag{15}$$

This criterion penalizes any deviation from −1, that is, the most negative value that $\mathcal{W}'_{i,j}$ can have. Here, $z_{99\%} = 2.326$ is the z-score of a Gaussian distribution at a 99% confidence level and $\sigma_{\mathcal{W},i,j}$ is the standard deviation of edge witness $\mathcal{W}_{i,j}$. These tests are conservative as they penalize any deviation from the ideal values. Moreover, these tests are most suitable for circuit cutting because the QPD may increase the variance $\sigma_{\mathcal{W}_{i,j}}$ of the measured witnesses. Therefore, the statistics of entanglement are detected only if the mean of a witness is sufficiently negative and its standard deviation is sufficiently small. An edge $(i,j) \in E$ fails the criteria if equation (14) or equation (15) is not satisfied. All edges in E, including the cut edges, pass the test based on $\mathcal{W}_{i,j}$ when implemented with LO and LOCC (Extended Data Table 2). However, some edges fail the test based on $\mathcal{W}'_{i,j}$ because of the lower noise robustness of $\mathcal{W}'_{i,j}$ compared with $\mathcal{W}_{i,j}$.

Circuit count for stabilizer measurements

Obtaining the bipartite entanglement witnesses requires measuring the expectation values of $\langle S_i\rangle$, $\langle S_j\rangle$ and $\langle S_iS_j\rangle$ of each edge $(i,j) \in E$. For the 103- and 134-node graphs presented in the main text, all 219- and 278-node and edge stabilizers, respectively, can be measured in $N_S = 7$ groups of commuting observables. To mitigate final measurement readout errors, we use twirled readout error extinction (TREX) with N_{TREX} samples[57]. When virtual gates are used with LO and LOCC, we require I_{LO} and I_{LOCC} more circuits, respectively. In this work, we fully enumerate the QPD. Furthermore, for LOCC, we mitigate the delay of the switch instruction with ZNE based on N_{ZNE} stretch factors. Therefore, the four types of experiments are executed with the following number of circuits.

- Swaps: N_SN_{TREX}
- Dropped edge: N_SN_{TREX}
- LO: $N_SN_{TREX}I_{LO}$
- LOCC: $N_SN_{TREX}I_{LOCC}N_{ZNE}$

In the experiments for the 103- and 134-node graph states, we use $N_{TREX} = 5$ and 3 TREX samples, respectively. Therefore, measuring the stabilizers without a QPD requires $N_S \times N_{TREX} = 35$ circuits for the 103-node graph. For LO and LOCC, measuring the stabilizers for the graphs in the main text requires 6^4 and 27^2 circuits, respectively. However, owing to the graph structure, each edge witness is only ever in the light cone of two cut gates at most. We may thus execute a total of $I_{LO} = 6^2$ and $I_{LOCC} = 27$ circuits for LO and LOCC, respectively, based on the light cone of the gates. For higher-weight observables, this corresponds to sampling the diagonal terms of a joint QPD. Therefore, measuring the stabilizers with LO requires $N_S \times N_{TREX} \times I_{LO} = 1{,}260$ circuits. For LOCC, we further perform error mitigation of the switch with $N_{ZNE} = 5$ stretch factors. We, therefore, execute $N_S \times N_{TREX} \times I_{LOCC} \times N_{ZNE} = 4{,}725$ circuits to measure the error-mitigated stabilizers needed to compute $\mathcal{W}_{i,j}$. Each circuit is executed with a total of 1,024 shots.

To reconstruct the value of the measured observables, we first merge the shots from the TREX samples. To do this, we flip the classical bits in the measured bit strings corresponding to measurements for which TREX prepended an X gate. These processed bit strings are then aggregated in a count dictionary with $1{,}024 \times N_{TREX}$ counts. Next, to obtain the value of a stabilizer, we identify which of the N_S measurement bases we need to use. The value of a stabilizer and its corresponding standard deviation are then obtained by resampling the corresponding $1{,}024 \times N_{TREX}$ counts. Here, we randomly select 10% of the shots to compute an expectation value. Ten such expectation values are averaged

and reported as the measured stabilizer value. The standard deviation of these 10 measurements is reported as the standard deviation of the stabilizer, shown as error bars in Fig. 2b. Finally, if the stabilizer is in the light cone of a virtual gate implemented with LOCC, we linearly fit the value of the stabilizer obtained at the $N_{ZNE} = 5$ switch stretch factors. This fit, shown in Extended Data Fig. 2d, enables us to report the stabilizer at the extrapolated zero-delay switch.

Data availability

The code to analyse the counts, reproduce the plots in this paper and produce the circuits for the cut Bell pairs are available on GitHub (https://github.com/eggerdj/cut_graph_state_data). The raw counts are unavailable on GitHub because of size constraints but are available upon reasonable request.

44. Cai, Z. et al. Quantum error mitigation. *Rev. Mod. Phys.* **95**, 045005 (2023).
45. Endo, S., Benjamin, S. C. & Li, Y. Practical quantum error mitigation for near-future applications. *Phys. Rev. X* **8**, 031027 (2018).
46. McKay, D. C., Wood, C. J., Sheldon, S., Chow, J. M. & Gambetta, J. M. Efficient *Z* gates for quantum computing. *Phys. Rev. A* **96**, 022330 (2017).
47. Tran, M. C. et al. Hierarchy of linear light cones with long-range interactions. *Phys. Rev. X* **10**, 031009 (2020).
48. Zettles, G., Willenborg, S., Johnson, B. R., Wack, A. & Allison, B. 26.2 Design considerations for superconducting quantum systems. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 65, pp. 1–3 (IEEE, 2022).
49. Cross, A. et al. OpenQASM 3: a broader and deeper quantum assembly language. *ACM Trans. Quantum Comput.* **3**, 1–50 (2022).
50. Vidal, G. & Tarrach, R. Robustness of entanglement. *Phys. Rev. A* **59**, 141–155 (1999).
51. Kraft, D. *A Software Package for Sequential Quadratic Programming* (DFVLR, 1988).
52. Gupta, R. S. et al. Probabilistic error cancellation for dynamic quantum circuits. *Phys. Rev. A* **109**, 062617 (2024).
53. Briegel, H. J. & Raussendorf, R. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.* **86**, 910–913 (2001).
54. Hein, M., Eisert, J. & Briegel, H. J. Multiparty entanglement in graph states. *Phys. Rev. A* **69**, 062311 (2004).
55. Jungnitsch, B., Moroder, T. & Gühne, O. Entanglement witnesses for graph states: general theory and examples. *Phys. Rev. A* **84**, 032310 (2011).
56. Tóth, G. & Gühne, O. Entanglement detection in the stabilizer formalism. *Phys. Rev. A* **72**, 022340 (2005).
57. van den Berg, E., Minev, Z. K. & Temme, K. Model-free readout-error mitigation for quantum expectation values. *Phys. Rev. A* **105**, 032620 (2022).

Acknowledgements We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team. We acknowledge B. Donovan, I. Hincks and K. Barton for circuit compilation and execution. We also thank D. Sutter, J. Orcutt, E. van den Berg, K. Temme, L. Bishop and P. Seidler for their discussions.

Author contributions A.C.V. gathered the LO data and studied the error mitigation of MCMs. C.T. studied the error mitigation of dynamic circuits and graph states. D.J.E. gathered the LOCC data. A.C.V., C.T. and D.J.E. analysed the circuit-cutting data. D.R. implemented the error suppression for dynamic circuits and optimized their execution on hardware. S.W. designed the cut Bell pairs. M.T. coordinated hardware resources and experiment execution. D.J.E. and M.T. designed the graph state experiments. D.J.E. designed the switch error mitigation. All authors discussed the results and improved and approved the paper.

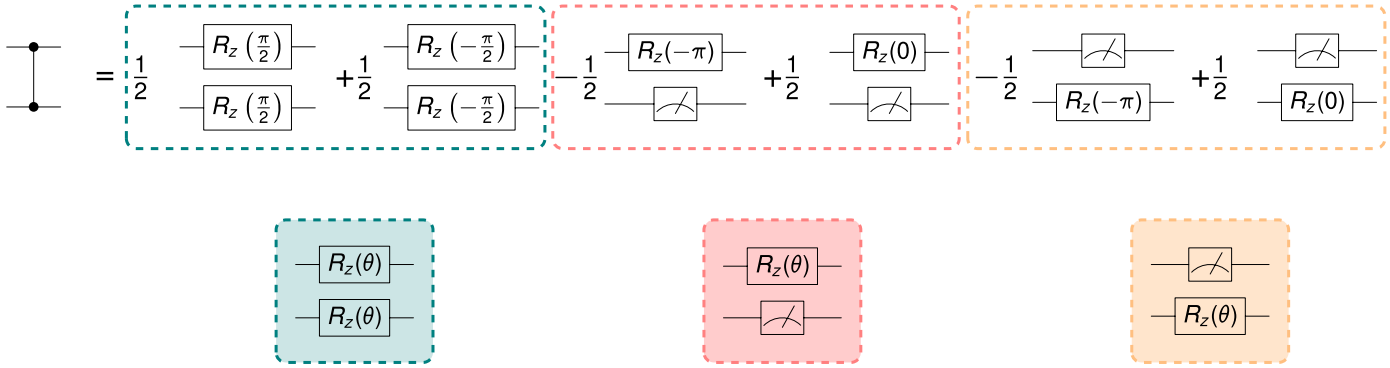
Competing interests A patent (application no. 18/523211) was filed on 29 November 2023 with listed inventors D.J.E., C.T., D.R., A.C.V., S.W. and M.T. The authors declare no other competing financial or non-financial interests.

Additional information

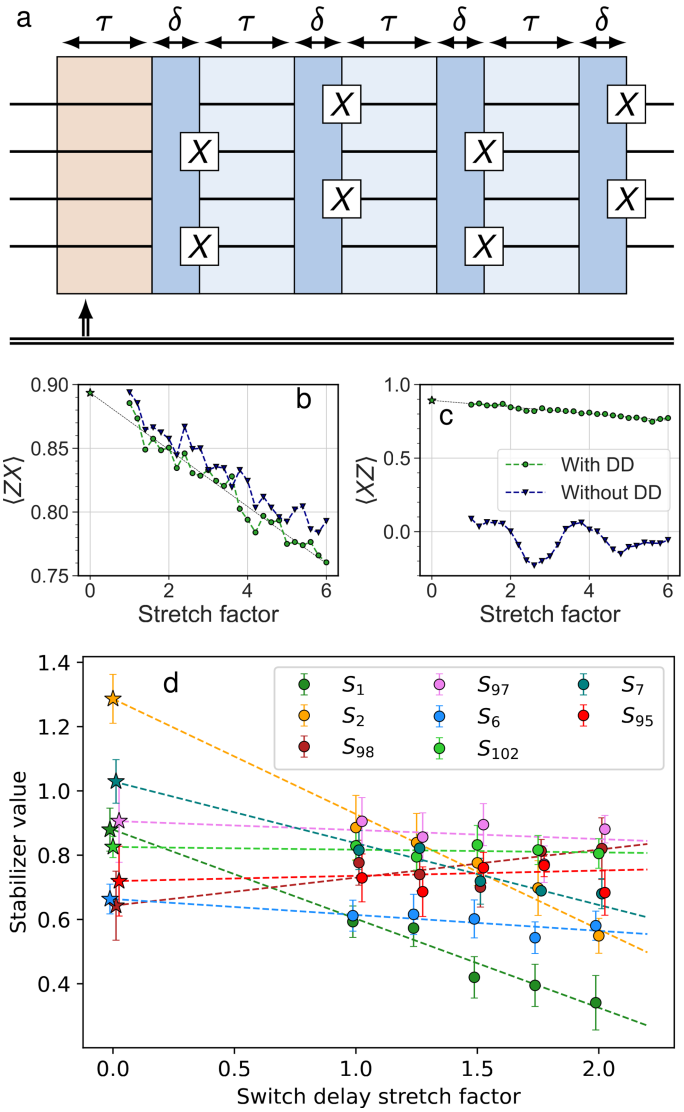
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-024-08178-2.

Correspondence and requests for materials should be addressed to Daniel J. Egger.

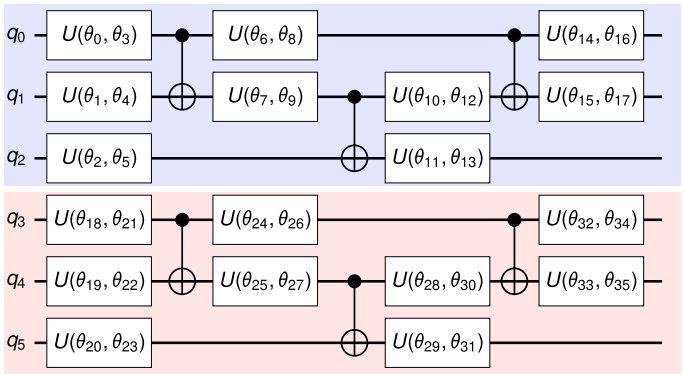
Peer review information *Nature* thanks the anonymous reviewers for their contribution to the peer review of this work. Peer reviewer reports are available.



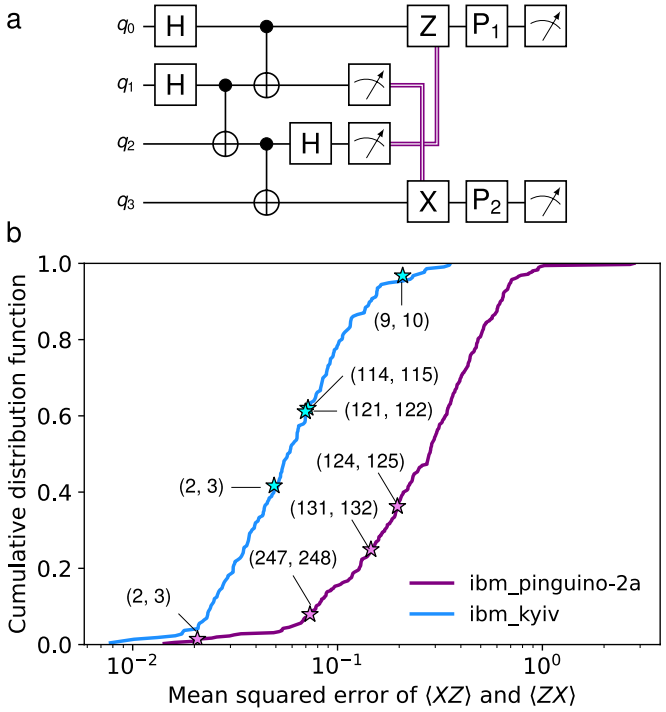
Extended Data Fig. 1 | LO decomposition of a CZ gate. A single CZ gate can be simulated through local operations by sampling from the shown QPD and applying classical post-processing to the results. Each of the six circuits has a sampling probability of $1/(2\gamma) = 1/6$. For the four circuits featuring mid-circuit measurements, the corresponding QPD coefficient is adjusted by a factor of +1 for outcome 0 and a factor of −1 for outcome 1. To optimize the execution, these six circuits are consolidated into three parametrized circuits to enable a parametric circuit execution. The green, red, and yellow circuits correspond to the three template circuits generated from cutting a single CZ gate with the LO protocol. Here, the presence or absence of a mid-circuit measurement changes the pulse-level payload which thus requires compilation.



Extended Data Fig. 2 | Zero-noise extrapolation of a switch. a, Implementation of a switch with DD. The conditional gates (not shown) are executed after the last DD X gate. The red delay of τ shows the duration in which no gates can be executed as the control electronics is busy, see Sec. VIC 1. The three additional delays of τ enable staggered DD. The four additional and variable delays of δ allow us to vary the duration of the switch for ZNE. **b**,**c**, The ZX and XZ correlators measured on *ibm_peekskill* as a function of the switch stretch factor c for a two-qubit graph state on $G = (\{0,1\}, \{(0,1)\})$. **d**, Example correlators of the 103 node graph extrapolated with ZNE.



Extended Data Fig. 3 | Quantum circuit of three cut Bell pairs. A sum over the right set of parameter vectors $\{\boldsymbol{\theta}\}$ results in three cut Bell pairs between qubit pairs (q_0, q_3), (q_1, q_4), and (q_2, q_5). The gate $U(\theta, \phi)$ corresponds to the gate sequence $\sqrt{X}R_z(\theta)\sqrt{X}R_z(\phi)$. The blue and red shaded regions correspond to the two disjoint portions of the quantum circuit.



Extended Data Fig. 4 | LOCC Bell pair benchmark. a, Quantum circuit that creates an uncut Bell pair on qubits (1, 2) and consume it in a teleportation circuit to create a Bell state on qubits (0, 3). **b**, Cumulative distribution function of the MSE of $\langle ZX \rangle$ and $\langle XZ \rangle$ for all groups of four linearly connected qubits on each device. The stars correspond to the qubits used in the 103- and 134-node graph states presented in the main text. The numbers in brackets indicate the qubit numbers corresponding to (q_1, q_2) in panel (a).

Extended Data Table 1 | Circuit structure and node error

	γ^{2n}	Nbr. CNOTs	Nbr. MCM	$\sum_{i\in V} \lvert S_i - 1\rvert$
Dropped edge	1	112	0	13.1
SWAPs	1	374	0	44.3
LOCC	49	128	8	8.9
LO	81	112	8/3	7.0

The circuits are transpiled to hardware-native CNOT gates. The number of MCMs for LO varies with the different circuits in the QPD. We, therefore, report the average number of MCMs.

Extended Data Table 2 | Witness tests

Graph	103 nodes		134 nodes	
Criteria	$\mathcal{W}_{i,j}$ (14)	$\mathcal{W}'_{i,j}$ (15)	$\mathcal{W}_{i,j}$ (14)	$\mathcal{W}'_{i,j}$ (15)
SWAP	70%	48%	n.a.	n.a.
Dropped-edge	89%	88%	92%	85%
LOCC	100%	100%	100%	87%
LO	100%	100%	100%	96%

Fraction of the edges in the graph state that passes the entanglement witness tests. For the dropped edge benchmark, we expect a pass rate of at most 88% and 92% for the 103- and 134-node graphs, respectively. The 89% measured pass rate of dropped edge for the graph state with 103 nodes exceeds this value because of a single edge that barely passes the test with $\mathcal{W}_{i,j} = -0.0917$ and $s_{\mathcal{W},i,j} = 0.0146$ due to measurement fluctuations. NA, not applicable.