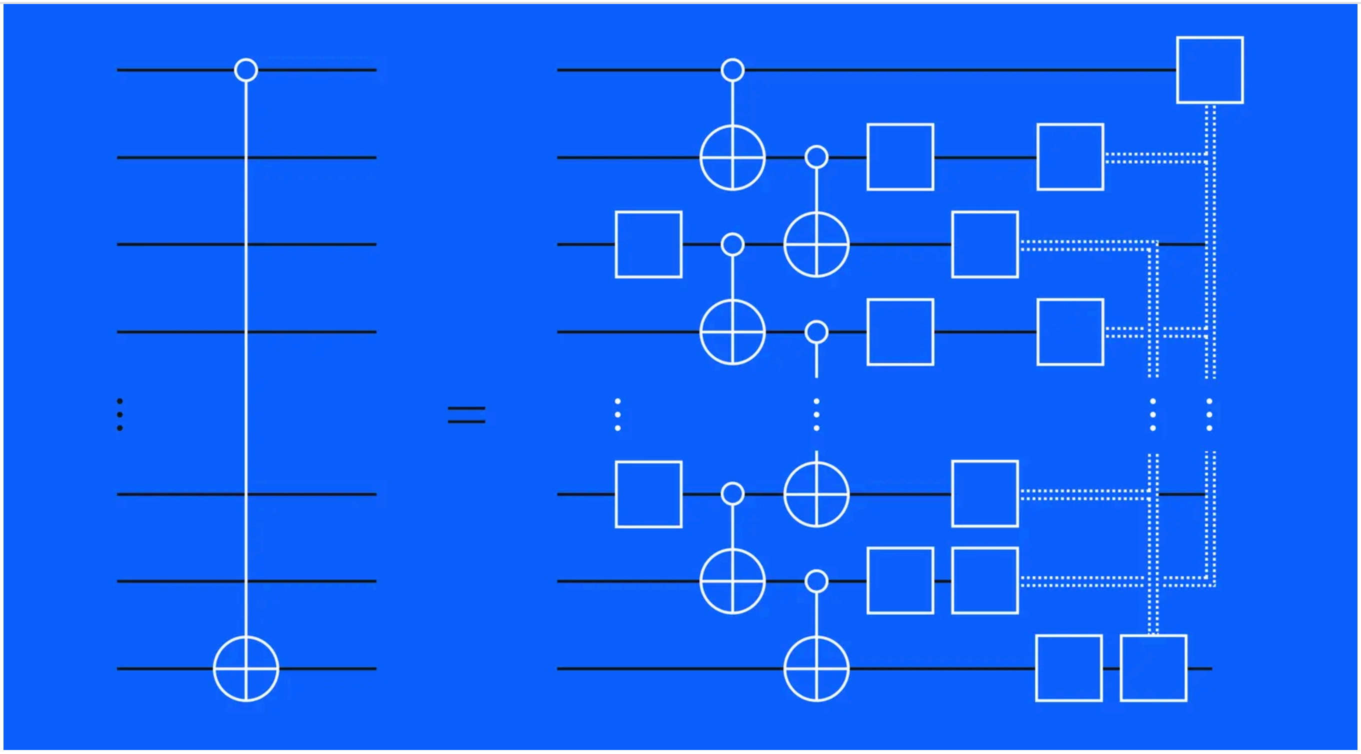
Using dynamic circuits to efficiently implement quantum states with long-range entanglement

Researchers from IBM Quantum™ implement an efficient new strategy for connecting distant qubits, overcoming the limited connectivity of current quantum chips.

IBM

Sign in to Platform ↗



Date

22 Jan 2025

Authors

Simone Cantori
Marcel Pfaffhauser
Elisa Bäumer
Fabio Scafirimuto
Robert Davis

Topics

Research

Share this blog

Blog Summary:

- Quantum entanglement is a phenomenon that allows quantum computers to process information in ways that are challenging or impossible for classical computers.
- However, superconducting quantum processors are limited in their ability to entangle qubits that are not directly connected to each other.

simulations on IBM Quantum hardware.

Dynamic circuits are powerful tools that give us the ability to measure qubits in the middle of a quantum circuit execution, and then perform classical logical operations within the circuit based on the outcome of those mid-circuit measurements. All of this occurs in the blink of an eye, before the circuit execution is complete, allowing us to take creative approaches to solving problems with quantum computation. These capabilities will be essential for the modular quantum computing architectures and quantum error correction protocols of the future. However, we still have a lot of work to do to explore their full potential.

In a recent blog post, we saw how IBM® researchers used dynamic circuits to share the statistics of entanglement over a classical communications channel, enabling the implementation of virtual two-qubit gates between qubits on separate quantum processors. However, the usefulness of dynamic circuits doesn't end there. A paper published in PRX Quantum by other IBM researchers shows how dynamic circuits can enable efficient long-range entanglement between qubits on the same chip using shallow circuits. The researchers behind this second experiment were able to implement their technique on IBM's superconducting quantum computers using up to 101 qubits.

Creating long range entanglement in a single QPU may not seem too far off from the sharing of entanglement between separate QPUs, but there are some key differences between the two methods. Where the latter method expands the scale of the circuits we can execute in terms of qubit count, the other greatly expands the level of complexity we can represent in our quantum circuits through qubit entanglement.

In this article, we'll explain how the IBM researchers uses dynamic circuits to create long-range entangling gates, and to prepare long-range entangled states in locally connected qubits. Along the way, we'll also explore the quantum computation.

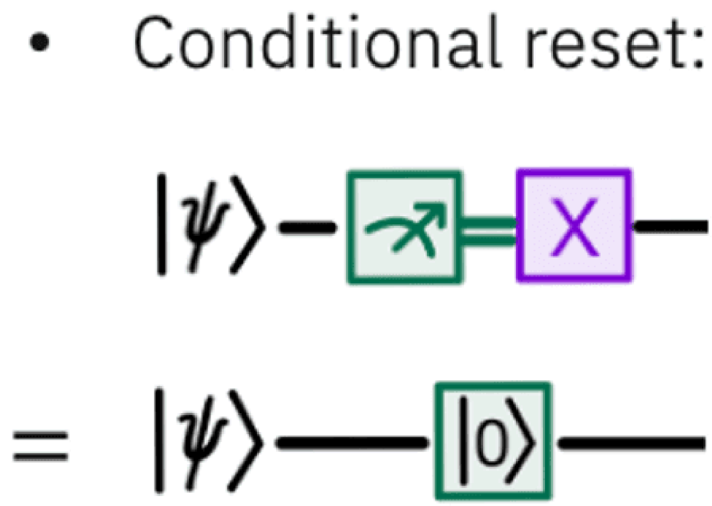
The power of dynamic circuits

Quantum computers have the potential to solve some computational problems more efficiently than any classical computing method. However, for many computing tasks, classical computation remains faster and more reliable. In fact, quantum computers rely on classical computation for both preparing quantum circuits and analyzing the results of circuit executions, and increasingly, quantum researchers are developing algorithms that exploit both quantum and classical methods in the middle of those circuit executions as well.

Algorithms powered by dynamic circuits are a promising and extremely versatile example of how quantum and classical methods can work together within a quantum circuit. Dynamic circuits use mid-circuit measurements to implement feed-forward operations—i.e., classical calculations based on the results of measurements performed on qubits in the circuit. These classical calculations determine subsequent gate applications as the circuit execution continues.

A simple example of this is the reset operation that resets a qubit to the $|0\rangle$ state. You might implement this by programming a dynamic circuit to measure a certain qubit and either apply an X gate if the measurement outcome is 1, or do nothing if the measurement outcome is 0.

What does the X gate do? Applying an X gate to a qubit in state $|1\rangle$ will change the qubit to state $|0\rangle$. This is also known as a “bit flip,” because it “flips” the qubit state.



The image to the left is taken from episode 153 of the Qiskit Seminar Series on the Qiskit YouTube channel, which features lead author Elisa Bäumer discussing the PRX Quantum paper. The image depicts the scheme of a reset operation in a dynamic circuit, where the application of the X gate is conditioned upon the result of the mid-circuit measurement.

The IBM researchers behind the PRX Quantum paper use dynamic circuits to generate long-range entanglement between qubits using shallow-depth quantum circuits. The methods they demonstrate in their paper serves to overcome the limited connectivity between qubits in superconducting quantum chips while also making the qubits more resilient against environmental noise.

gates

One of the fundamental properties of quantum computers that gives them advantage over classical computers is their ability to create quantum entanglement between qubits. Quantum entanglement is achieved when the state of one entangled qubit influences the state of the other.

The controlled-not (CNOT) gate is one of the most common two-qubit quantum gates we use to generate entanglement between qubits. A CNOT gate always involves two qubits—a control qubit and a target qubit. If the control qubit is in the $|1\rangle$ state, the CNOT gate flips the target qubit. If the control qubit is in any other state, nothing happens. We represent this in Bra-ket notation as $CNOT(A|00\rangle + B|10\rangle) = A|00\rangle + B|11\rangle$.

Today's superconducting quantum chips have limited connectivity between qubits. For example, IBM Quantum chips are all limited to nearest-neighbor connectivity, meaning they are only connected to qubits immediately adjacent to them. However, quantum computing applications like quantum error correction and quantum simulation often require entanglement between distant qubits. The quantum community has developed methods for implementing these long-range CNOT gates

Current quantum chips have limited qubit connectivity, so you can't directly apply a CNOT gate between every pair of qubits. If you want to implement a CNOT gate between 2 qubits that are not connected to each other, you can use swap gates. As the name suggests, you can use swap gates to "swap" the position of qubits. For instance, you can use swap gates to implement a CNOT gate between two qubits, one in position A and the other in position C. They are not directly connected but share a connection with a third qubit in position B. First, you can use a swap gate between the qubit in position A and the qubit in B, then perform the CNOT operation between the qubit in B and the qubit in C, and finally apply another swap gate between the qubit in position A and the one in B to restore the original configuration. One problem with using swap gates is that they are implemented using 3 CNOT gates. CNOT are one of the main sources of noise and complexity in a quantum circuit, so we try to minimize how many of them we use.

the current technology.

The figure below illustrates how we would implement a long-range CNOT gate using both a traditional unitary circuit and a dynamic circuit incorporating classical logic.

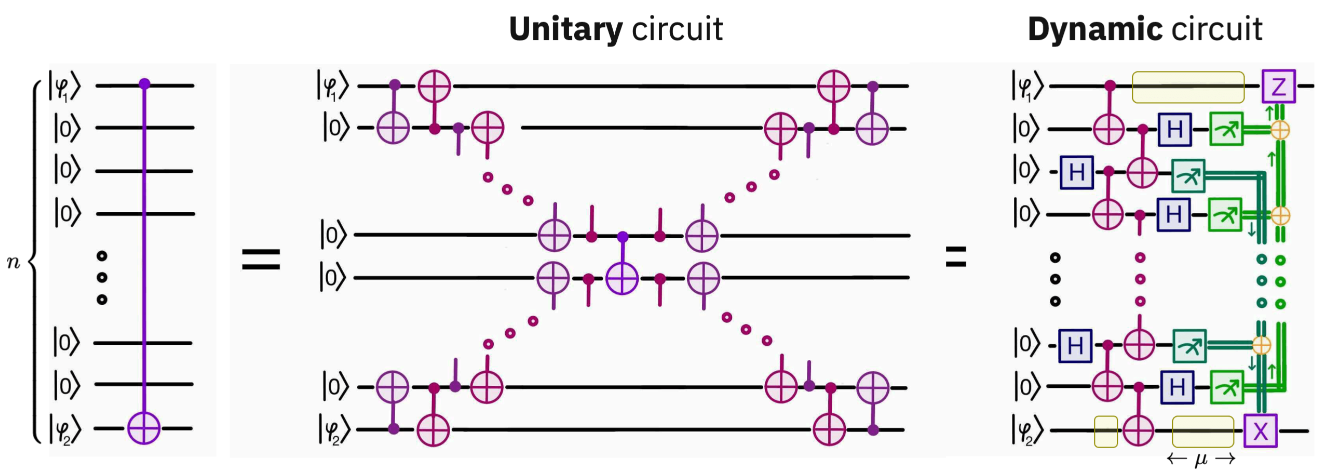


Image caption: Fig.1a of the paper. On the far left, we see a long-range CNOT gate. The diagram in the middle shows the equivalent quantum circuit we must implement to create the long-range CNOT gate in a quantum chip with only nearest-neighbor connections. The right shows the scheme of the dynamic circuit that implements the long-range CNOT gate.

To create a long-range CNOT gate using a normal unitary circuit on a chip with only nearest-neighbor connectivity, we must apply a linear number of CNOT gates—one for each of the qubit pairs situated between the main control and target qubits of the long-range CNOT pairing. This allows us to essentially send the entanglement across the chip like a game of “telephone,” where friends stand in a line and whisper a secret message from one person to the next.

person. Errors are introduced as the participants whisper the message from one person to the next, and the same is true for long-range CNOT gates. The circuit is often affected by noise introduced through the imperfect implementation of CNOT gates between neighboring qubits in the hardware. The chances of the circuit being negatively impacted by noise only increase as we add more qubits between the control and target of our long-range CNOT (i.e., as we increase the depth of the circuit).

How to build the long-range CNOT with dynamic circuits

Using the method described above, we need *4N+1* CNOT gates to implement our long-range CNOT, where *N* represents the number of ancilla qubits between the long-range CNOT's control and target. That's a lot of CNOT gates, especially given that every two-qubit gate added to a quantum circuit significantly increases the circuit's overall complexity.

To overcome this problem, the authors of the research paper propose an equivalent method that uses dynamic circuit capabilities to implement long-range CNOT gates with comparatively shallow circuits. Their method requires only *N+1* CNOTs—a significant reduction in complexity. Here's what a long-range CNOT gate looks like using dynamic circuits:

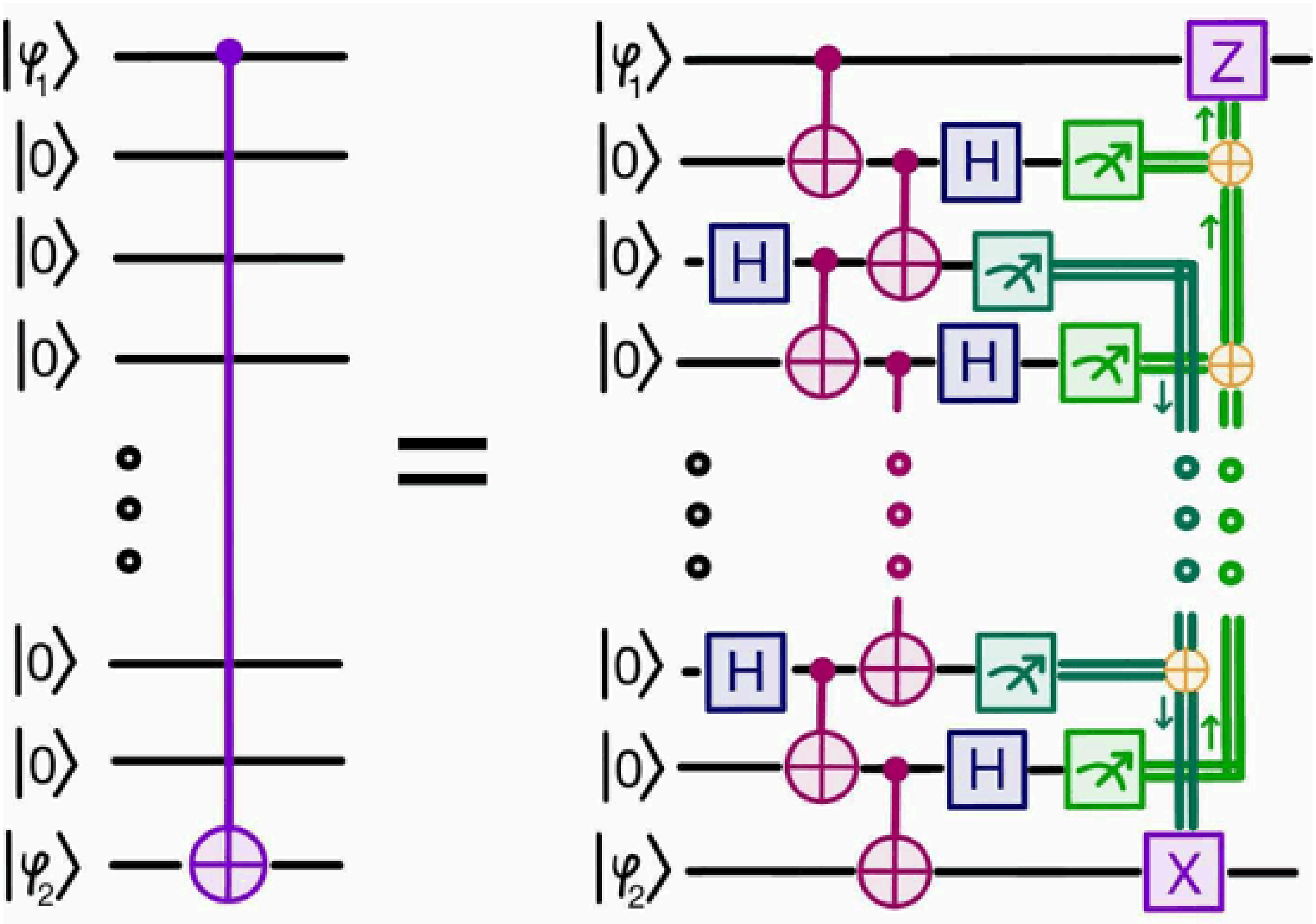


Fig.4 of the paper. Scheme of the procedure to obtain the long-range CNOT with a dynamic circuit.

Let’s take a moment to break down what’s happening in this image.

Regardless of whether you’re building your long-range CNOT gate with a dynamic circuit or a traditional unitary circuit, you will first need to entangle the long-range gates target and control qubits with all the qubits situated between them. Then, you must disentangle all the qubits in the middle, leaving entanglement only between the first and last qubits—i.e., the control and target of the long-range gate. At the end of the circuit, we use the measurement results of the ancilla qubits between the long-range intended.

In the previous section, we saw how this works in a normal unitary circuit. Now, let's take a look at how it would work with a dynamic circuit. We'll do so using a small 5-qubit example, where our goal is to implement a CNOT gate between the first and last qubit. In this case, the final quantum state that we want to obtain between first and last qubit is a bell state *|psi⟩ =1/sqrt(2)(|00⟩ + |11⟩)*.

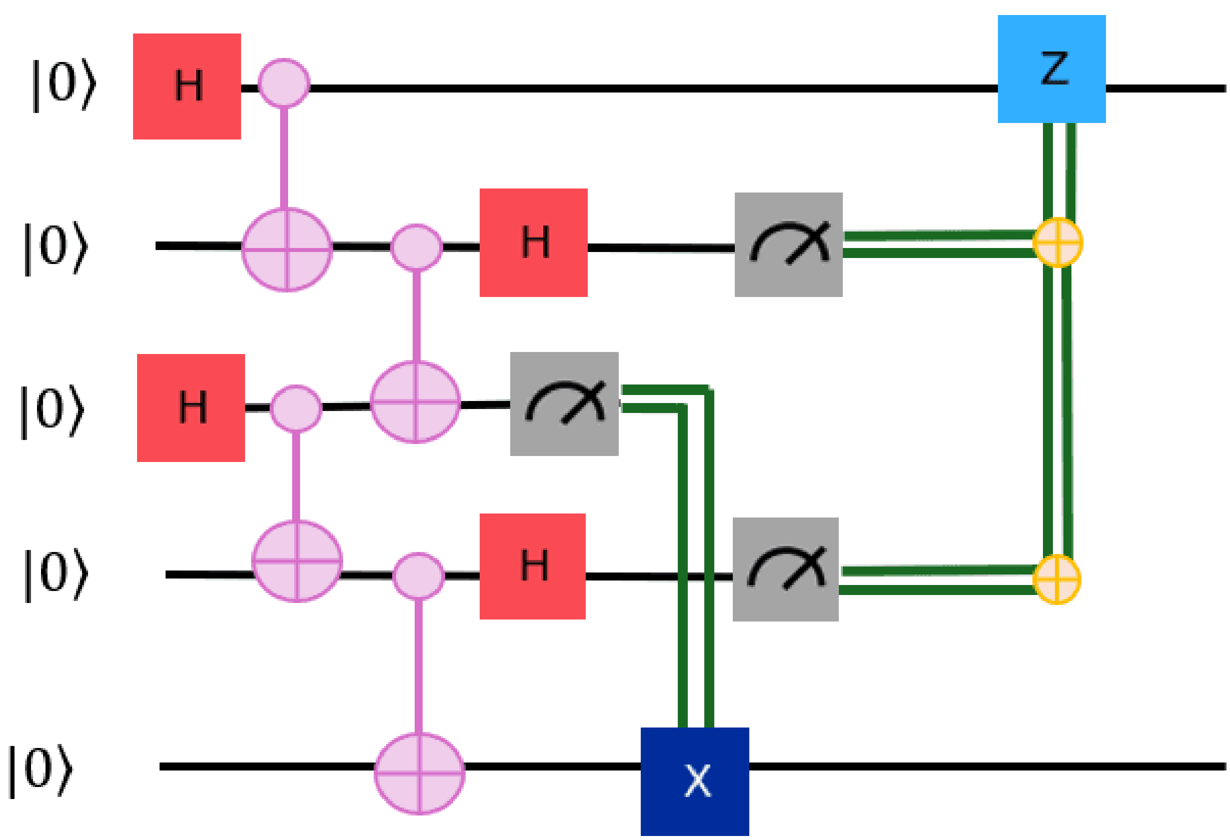


Diagram of a 5 qubit dynamic circuit implementation of a long-range CNOT between the first and last qubits. Image credit: Simone Cantori.

GHZ state for the last 3 qubits.

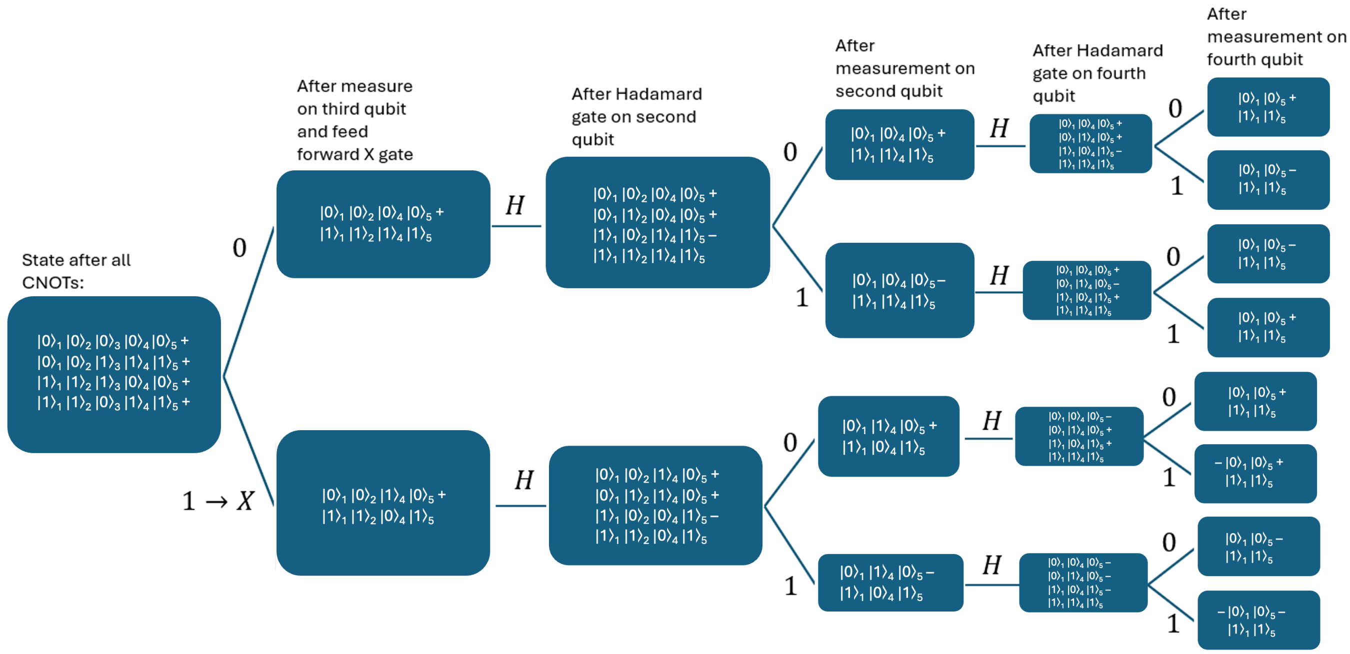
> **What are the Bell and GHZ states?** Bell state and GHZ are entangled quantum states. Their mathematical representations in this case are $|Bell\rangle=1/sqrt(2)(|00\rangle + |11\rangle)$ and $|GHZ\rangle=1/sqrt(2)(|000\rangle + |111\rangle)$.

Then, we'll need to entangle these two quantum states using another CNOT implemented between the second and third qubit. Since our final goal is to obtain a Bell state for the first and the last qubit, we have to disentangle and reset the ancilla qubits in the middle of the circuit.

The third qubit is entangled with and stores the parity of the others. This means that, after we measure it, we must ensure that the first and last qubit are both in a superposition of $|00\rangle$ and $|11\rangle$. To achieve this result, we implement a feed-forward X gate to the last qubit according to the measurement outcome.

From there, we apply Hadamard gates to the second and fourth qubits right before taking the mid-circuit measurement. However, this operation will change the phase of some elements in your quantum state from positive (+) to negative (-). We can retrieve the correct phase by applying a feed-forward Z gate on the first qubit immediately after the mid-circuit measurement. You can see a more formal representation of this procedure in the image below:

IBM Sign in to Platform ↗



Binary tree to represent the quantum state after each step of the 5-qubits example. The final possible quantum states are the ones before the feed-forward Z gate. In fact, you can see that the _|00⟩_ elements have a different sign than the _|11⟩_ ones only when one of the two measurements is 1, so when the feed forward Z will be applied. Image credit: Simone Cantori.

You can also see the intermediate steps for implementing a long-range CNOT gate with dynamic circuits in the animation seen here. (Video credit: Simone Cantori.)

It's worth noting that we could also obtain the same result using classical post-processing methods. In this case, rather than applying quantum gates determined by the results of the mid-circuit measurement (e.g. the final X and Z gates at the end of the dynamic circuit protocol), we would just end the circuit run at the measurement step and apply classical operations from there. With this post-processing approach, the results are not influenced by errors from the classical feed forward operations that take place in a to be at the end of your circuit. If the long-range CNOT gate you've created is part of a larger algorithm that requires the application of additional gates after the CNOT, the post-processing method won't work, since the rest of the computation depends on the quantum state we create after the implementation of the CNOT.

The dynamic circuit implementation of the long-range CNOT requires *3N* fewer CNOT gates than the unitary circuit version. However, it also requires *N* mid-circuit measurements. Given that these measurements are less prone to noise than *3N* CNOT gates, dynamic circuits appear to be a much more reliable alternative to the standard unitary circuit approach.

The IBM researchers demonstrate this advantage using the current fleet of IBM Quantum computers by calculating the fidelity of the implemented quantum gate.

IBM Sign in to Platform

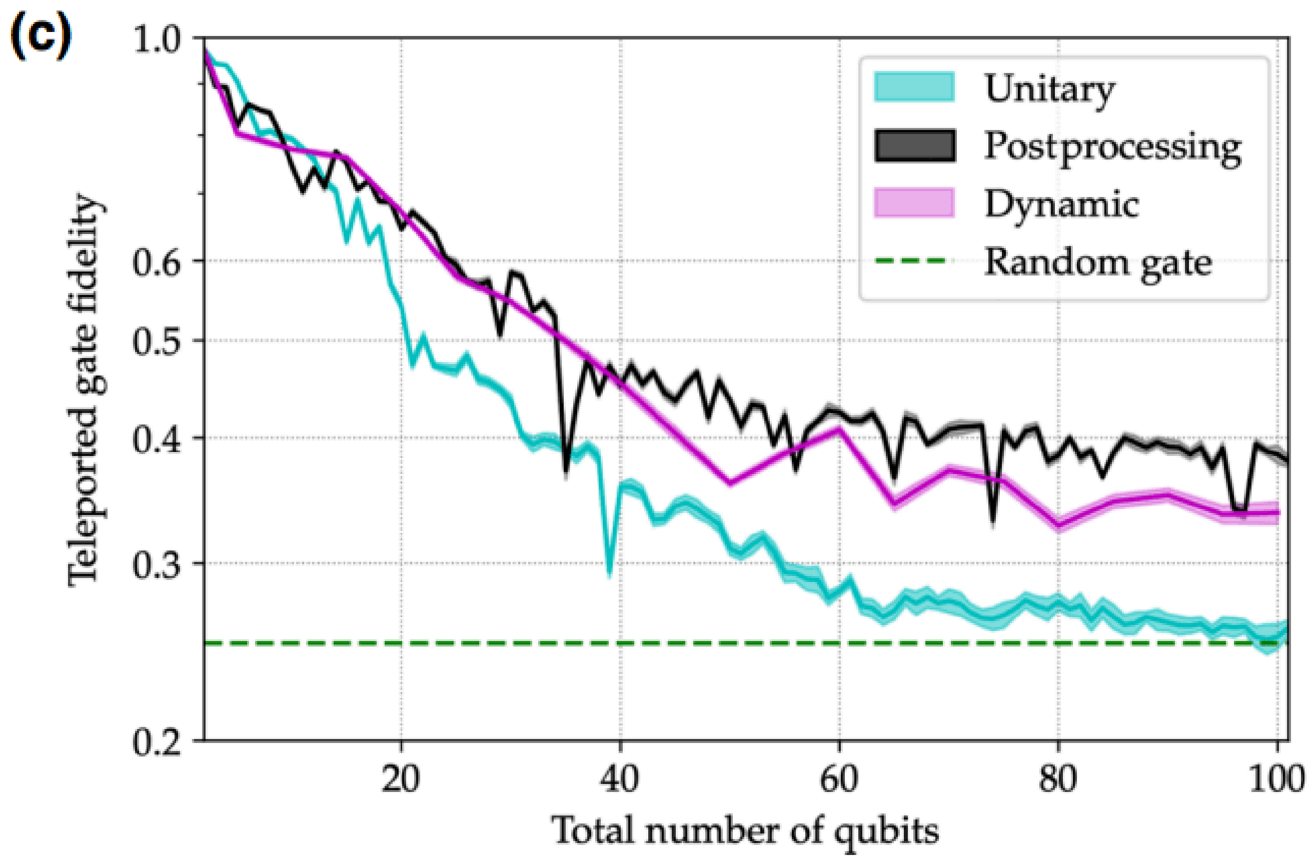


Fig.1c of the paper. Gate fidelity as a function of the number of qubits. The unitary circuit converges to the random gate fidelity (0.25) much faster than the dynamic circuit implementation.

Pushing long-range entanglement even further

In their paper, the IBM researchers show that we can use dynamic circuits to do much more than create individual long-range two-qubit gates. For instance, we can also use them to prepare entire quantum states with multiple long-range entanglements. As an example of this, let's look at the

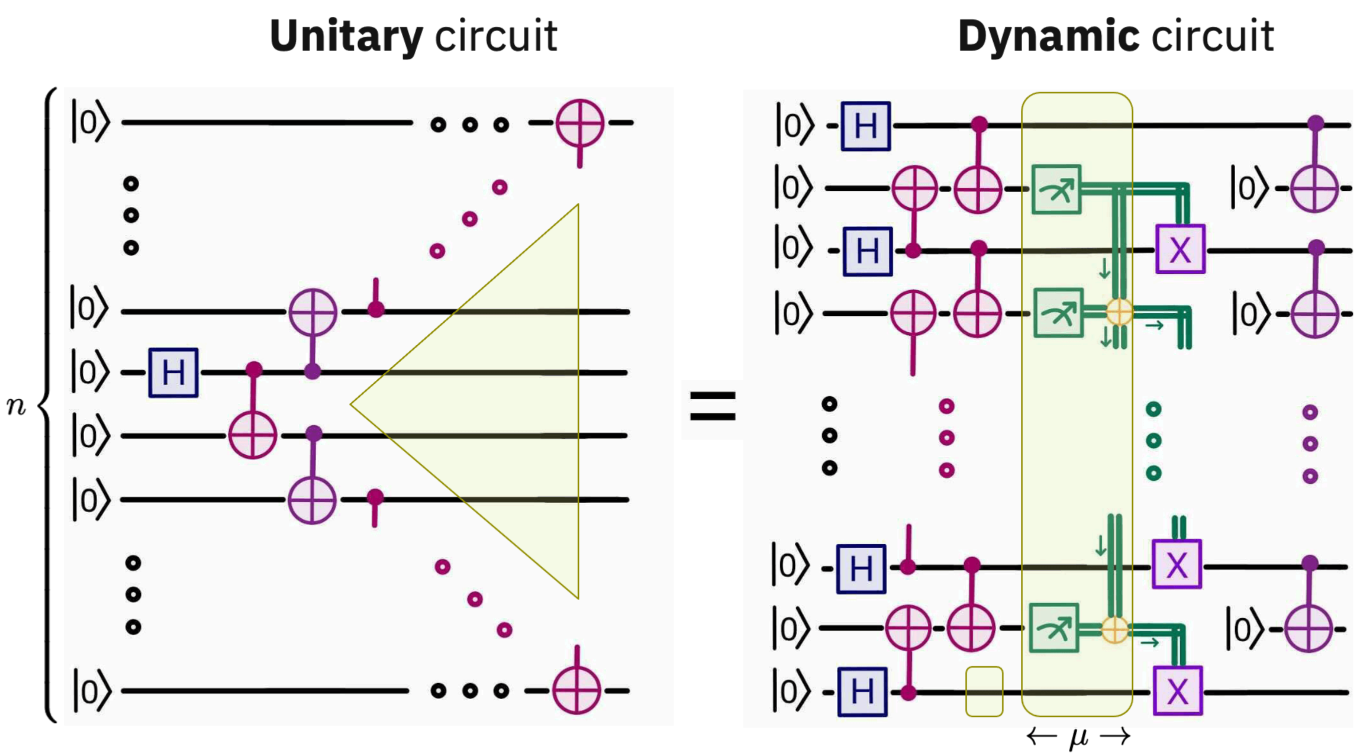


Fig.2b of the paper. Left side shows the unitary circuit implementation of a GHZ state. Right side shows the an equivalent scheme for implementing the same GHZ state using a dynamic circuit.

GHZ states are entangled quantum states involving three or more qubits. Their general representation is *|GHZ⟩=1/sqrt(2)(|00...0⟩ + |11...1⟩)*. One can see this as a generalization of the *1/sqrt(2)(|00⟩ + |11⟩* Bell state for more than two (i.e. "many") qubits.

The unitary circuit version of this GHZ state implementation requires *n-1* total CNOT gates, and the number of two-qubit gate layers scales linearly with the number of qubits.

qubits at the same time. The number of two-qubit gate layers is also called two-qubit gate depth.

The method for preparing a GHZ state with a dynamic circuit is similar to the method we for creating a long-range CNOT gate. First, you entangle pairs of qubits implementing Bell states. Then, you connect these pairs with an additional layer of CNOT gates, and you measure the target qubits of these CNOT gates.

Similar to the long-range CNOT gate, you can use feed-forward X gates after the measurements to avoid unwanted bitflips, and to make sure all entangled qubits are perfectly correlated. After that, you reset the measured qubits and apply a final layer of CNOT gates to obtain the GHZ state.

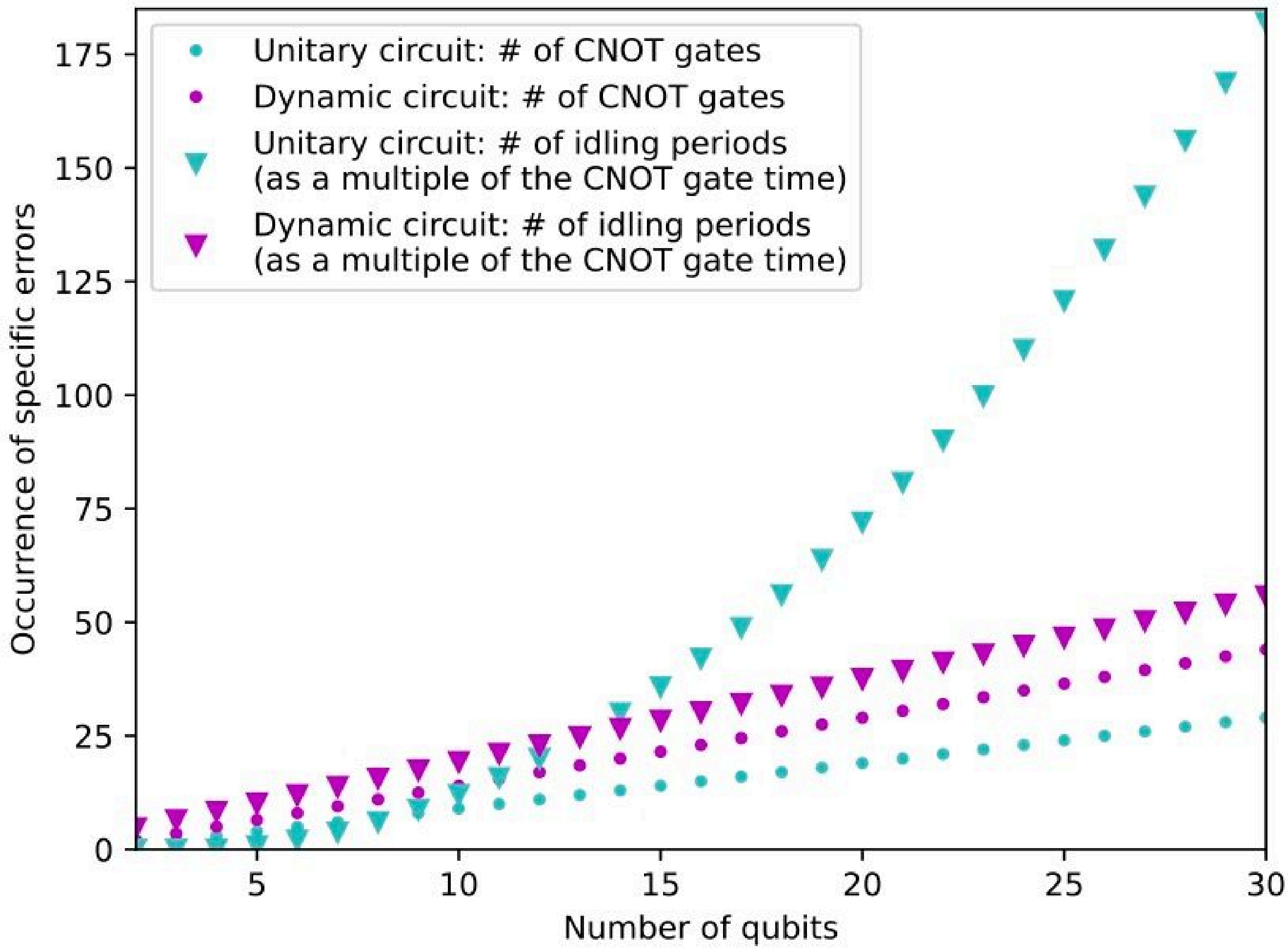
The depth of the dynamic circuit remains constant, but it is implemented using $3n/2 - 1$ total CNOT gates, and $n/2 - 1$ mid-circuit measurements. Although the dynamic circuit requires more CNOT gates and mid-circuit measurements, both of which are affected by noise, there still exists a regime in which implementing the dynamic circuit is more convenient than implementing the standard one. This is due to the idle time error

What is idle time error? In a two-qubit gate layer, it can happen that not all qubits are involved in the quantum gates. When this happens, the unused qubits are exposed to errors coming from interaction with the environment.

.

In fact, because the number of two-qubit gate layers scales linearly with the number of qubits, you get a quadratic scaling of the idle time error (approximately *n* qubits * *n* layers). This means that for large *n*, the impact of idle time error could be more significant than the impact of the CNOT gates and mid-circuit measurements. The threshold for defining the regime where the dynamic circuit is better than the unitary circuit also depends on the intensity of the noise acting on the different operations.

IBM Sign in to Platform



This graph shows the occurrence of two different sources of error for increasing number of qubits. The circles demonstrate how the number of CNOT gates increases as the number of qubits in the system grows, while the triangles represent the increasing idle time error. For dynamic circuits, we can see that both sources of error scale linearly. For unitary circuits, the number of CNOT gates scales linearly as well and is even below the number observed in dynamic circuits. However, errors caused by idle time increase quadratically, meaning that the total error of the unitary circuit quickly surpasses that of dynamic circuits.

In their experiments, the IBM researchers demonstrated how the use of dynamic circuits can enable higher fidelities on up to 101 qubits of a large-scale superconducting quantum processor. Their experiments show that dynamic circuits offer a promising solution for overcoming the connectivity limitations of large-scale noisy quantum hardware. For more on this

Quantum Platform to start experimenting with quantum computers today.

IBM Quantum: Tomorrow's computing today

Quantum starts here ↗

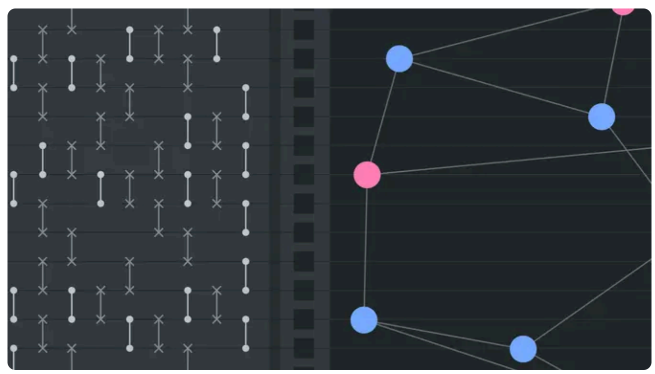
Keep exploring

View all blogs →

Melting magnets with utility-scale quantum simulation: Researchers explore condensed-matter physic...

12 Feb 2025 • Edward H. Chen, Robert Davis

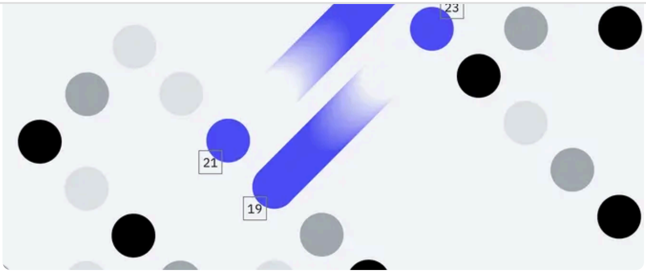
Research Use Cases



Exploring the potential for quantum advantage in mathematical optimization

5 Feb 2025 • Stefan Woerner, Ryan Mandelbaum, Robert Davis

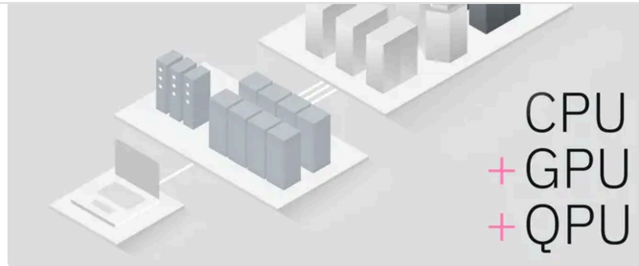
Community Research



Dynamic circuits enable essential circuit cutting methods for quantum-centric supercomputing

11 Dec 2024 • Daniel J. Egger, Maika Takita, Robert Davis

Error Correction & Mitigation | Research | Systems



Demonstrating a true realization of quantum-centric supercomputing

20 Nov 2024 • Ryan Mandelbaum, Iskandar Sitdikov

Research | Systems | Use Cases