Extractors: Scalable Quantum Computers of the Future

Zhiyang (Sunny) He[1], based on joint works with Alexander Cowtan[2], Dominic J. Williamson[3], and Theodore J. Yoder[3].

[1]MIT, [2]University of Oxford, [3]IBM Quantum

QMIT

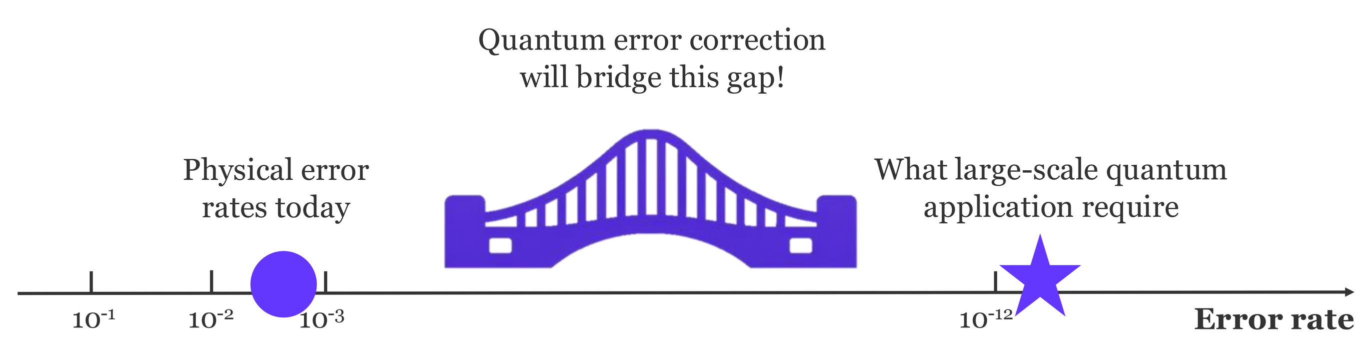
MIT Quantum Initiative

Quantum Computers: Fragile Yet Resilient

Quantum computing promises to be a revolutionary technology with classically intractable applications in cryptography, chemistry, and other disciplines. The future of computing will be redefined by large-scale, distributed networks of quantum computers.

Harvard IBM Google

However, current forms of quantum information are fragile: qubits are impacted by noise constantly. To run large-scale applications, we need to build noise-resilient systems with Quantum Error Correction.

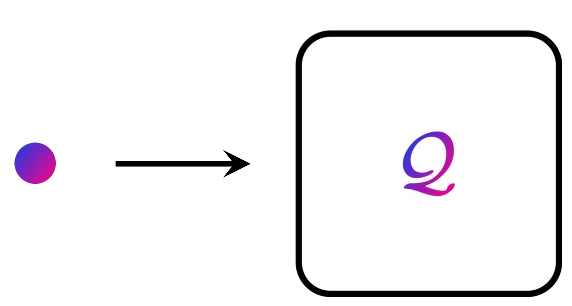


QEC is expensive: to factor a 2048-bit integer, we need thousands of noise-resilient qubits, which cost us million(s) of noisy qubits to simulate. How can we reduce this daunting cost and build efficient, scalable quantum computers for the future?

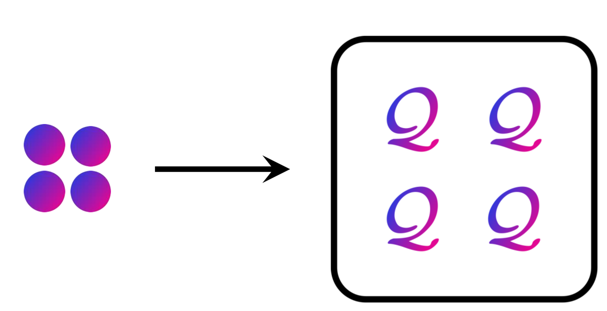
Quantum Error Correction: Store Few Into Many

The gist of QEC is to store the information of a few noise-resilient qubits, which we call logical qubits, into a larger number of noisy qubits, which we call physical qubits. Different quantum codes store information with different costs.

- Surface Code stores 1 logical qubit into a block of $2d^2$ many physical qubits, where d is a noise-resilience parameter.



- LDPC Codes store k logical qubits into a block of $c \cdot k$ many physical qubits, where c is between 10 to 50. This is independent of d!



For factoring, we typically set $d = 25$. Then…

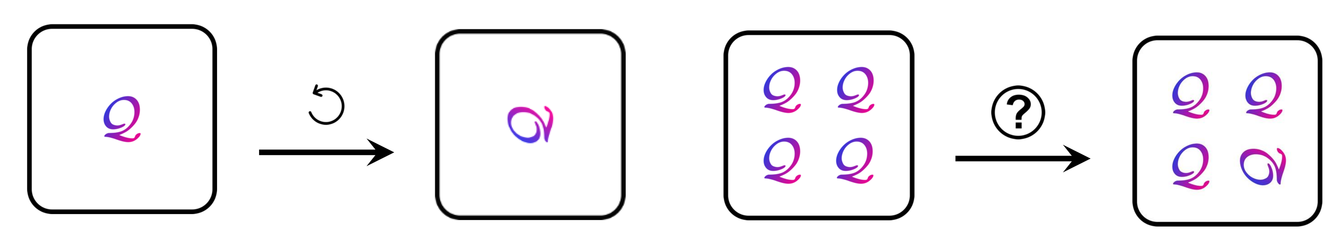
- Surface Code uses about 1350 physical qubits per logical qubit; This gives the million qubit estimate earlier.
- LDPC Codes uses about 10~50 physical qubits per logical qubit.

So using LDPC codes gives a clear, practical advantage… or does it?

Challenge: Computation on Stored Information

It's not enough to store information, we need to compute with them! The computation process needs to be noise-resilient as well.

Analogy: A quantum code is like a box, and logical qubits are colorful balls enclosed inside. To compute, we'd like to rotate the balls.

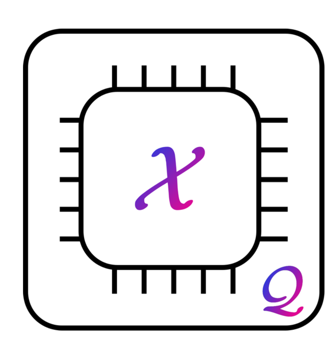


Surface Code: If there's only one ball inside, we can rotate the box. LDPC Codes: How can we rotate one ball without touching others?

This is a long-standing and well-studied challenge: LDPC codes are more efficient and scalable as memories, but harder to compute on.

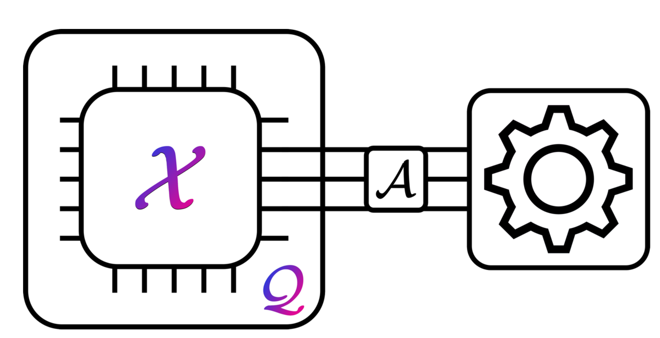
Extractors: Turning Memories Into Processors

Our solution: a powerful, auxiliary module that opens the box, extracts target logical qubits, rotate and put them back, then reseal the box. We call it an Extractor.

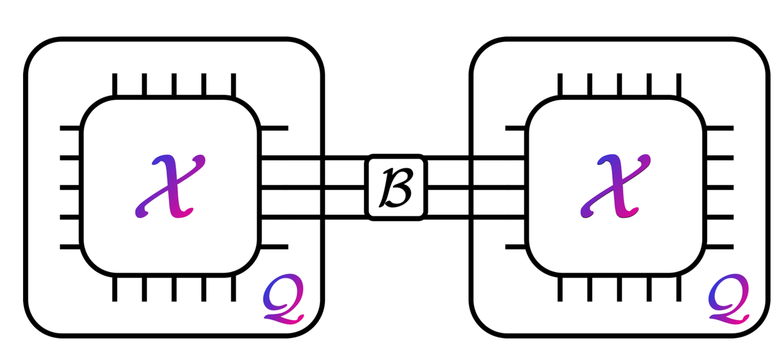


On top of any quantum code Q, we can build an extractor X. This design turns a noise-resilient quantum memory into a noise-resilient quantum processor. We call it an extractor-augmented computational (EAC) block.

To perform universal quantum computation (all possible angles of rotations), EAC blocks consume a type of resource called magic states. Magic states are used in almost all quantum computer architectures, including those built with the Surface Code.



An EAC block connected to a magic state factory has complete, noise-resilient control over stored quantum information.



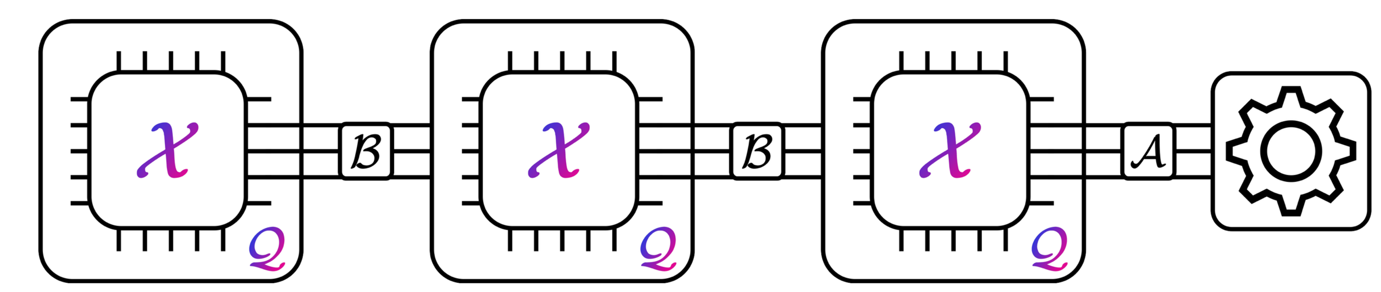
Multiple EAC blocks can be connected together to compute on a larger workspace.

In short: Extractor is all you need! It is a general, one-stop solution to the long-standing challenge of computing on efficient memories.

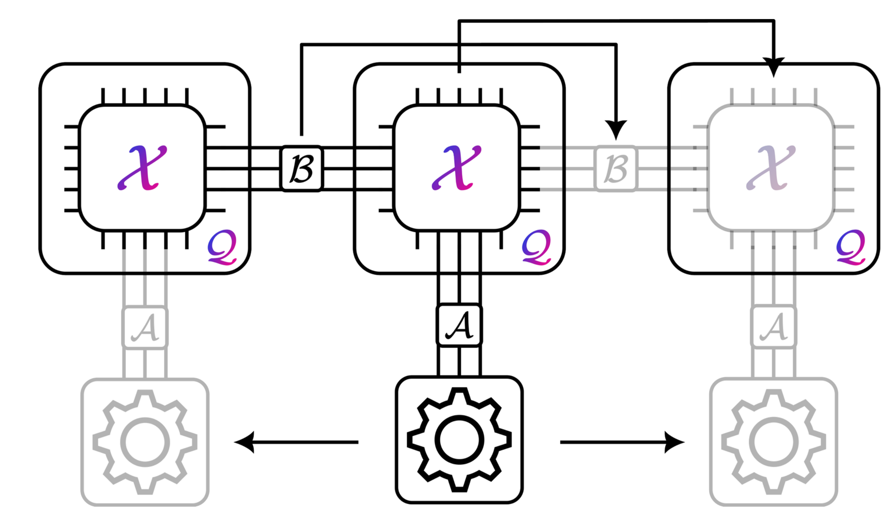
FAQ: What's the price tag? How many qubits do extractors cost? Answer: About the same size as the code/memory, but can be smaller! On IBM's 288-qubit bivariate bicycle code, we can build a 103-qubit partial extractor (arXiv 2407.18393). This is later improved to a 90-qubit design, now called a logical processing unit (arXiv 2506.03094).

Extractor Architectures: An Adaptable Blueprint

By connecting EAC blocks and magic state factories together, we can design different Extractor Architectures. E.g., we can build a chain of EAC blocks with one factory at the end. This is a theoretical model for IBM's bicycle architecture (arXiv 2506.03094).



Broadly, extractor architectures can be designed and built for many different hardware platforms/qubits, from fixed superconducting chips to mobile neutral atom arrays.



In a mobile neutral atom array, the many components of an extractor architecture can all be mobile.

Extractor architectures are general purpose quantum computers. They can be built with any quantum code or memory, and preserve the scalability advantage of efficient memories. With specific applications or hardware constraints in mind, the design of extractor architectures can be highly optimized. This is an adaptable blueprint for future, scalable, noise-resilient quantum computers.

Conclusion and Outlook

Building a large-scale quantum computer is an epochal scientific and technological endeavor of our time. While three decades of research have carefully studied and optimized Surface Code architectures, recent hardware progress enables us to use more complex LDPC codes to reduce the daunting space-cost of QEC. With the extractor architectures, we now have a theoretically general and practically promising blueprint to design scalable LDPC quantum computers of the future. In the next three decades of QEC, we expect extractor architectures to be designed, optimized, and built, for applications in factoring and beyond. Our journey has just begun, both challenges and discoveries await ahead.

Zhiyang He is a graduate student at MIT advised by Peter Shor and Anand Natarajan. This work (arXiv 2503.10390) builds upon research by IBM and collaborators. If you'd like to reach me, please email szhe@mit.edu. This poster is available on my website at this QR code. Talks are available online.

iQuISE MIT MIT Center for QuantumEngineering