~~Surface Code (Toric Code)~~ Quantum Computing

And

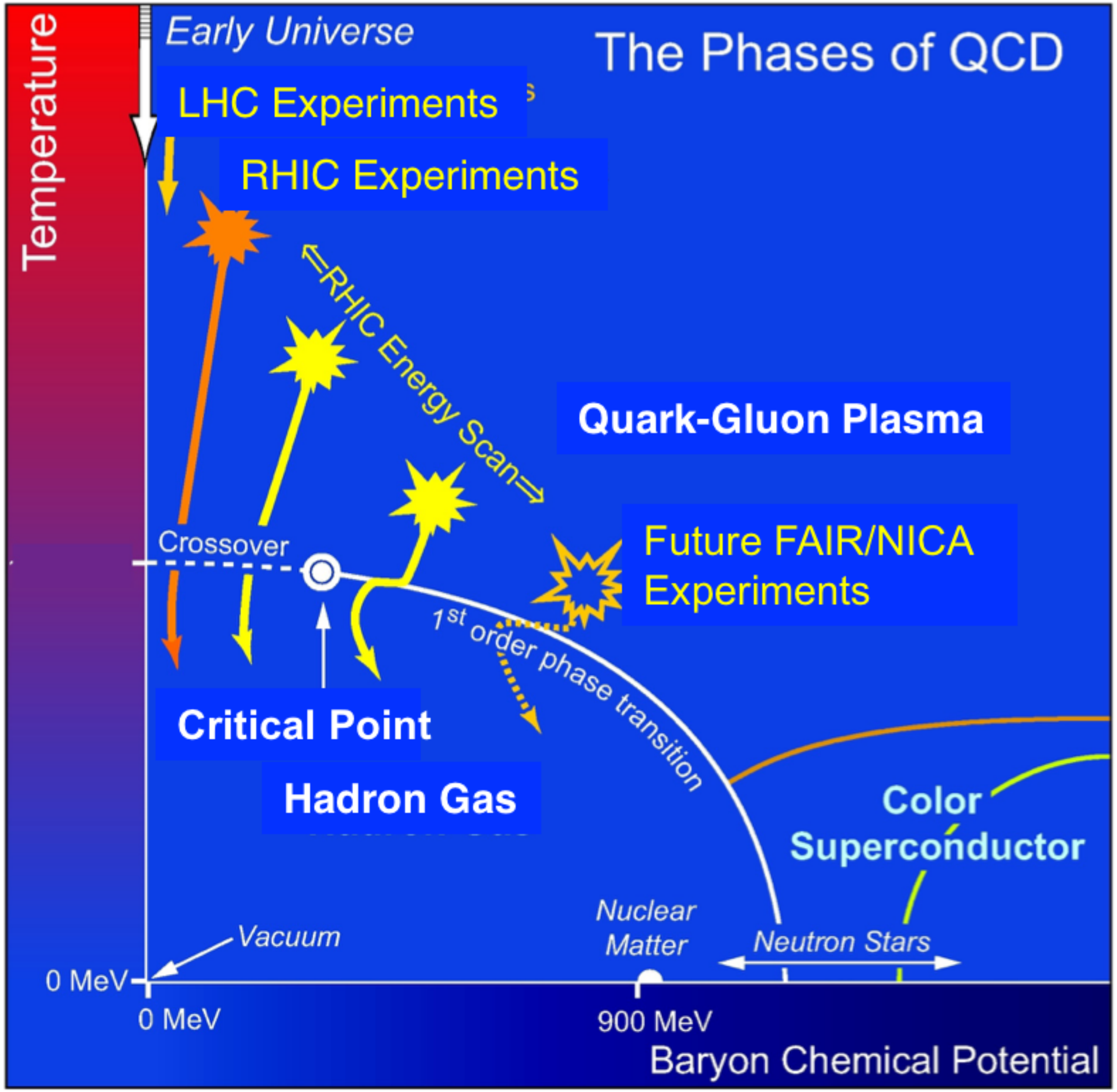
3−Dimensional Quenched $Z(2) \times Z(2)$ Gauge Theory

Seyong Kim
Sejong University

In collaboration with M. Mueller (IQI, RWTH), M. Rispler (IQI, RWTH) and D. Vodola (BASF)

Work in progress

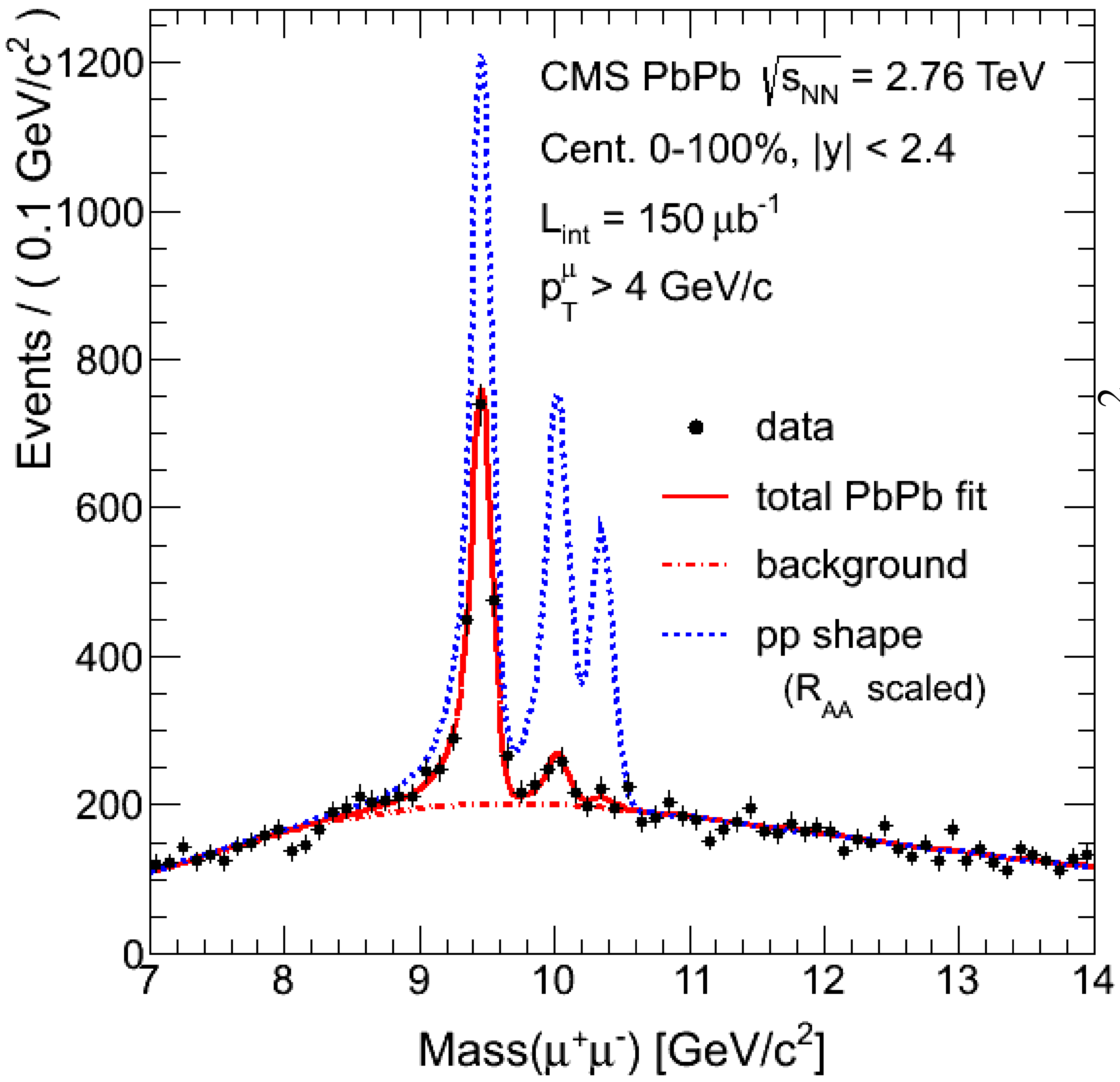
Motivation



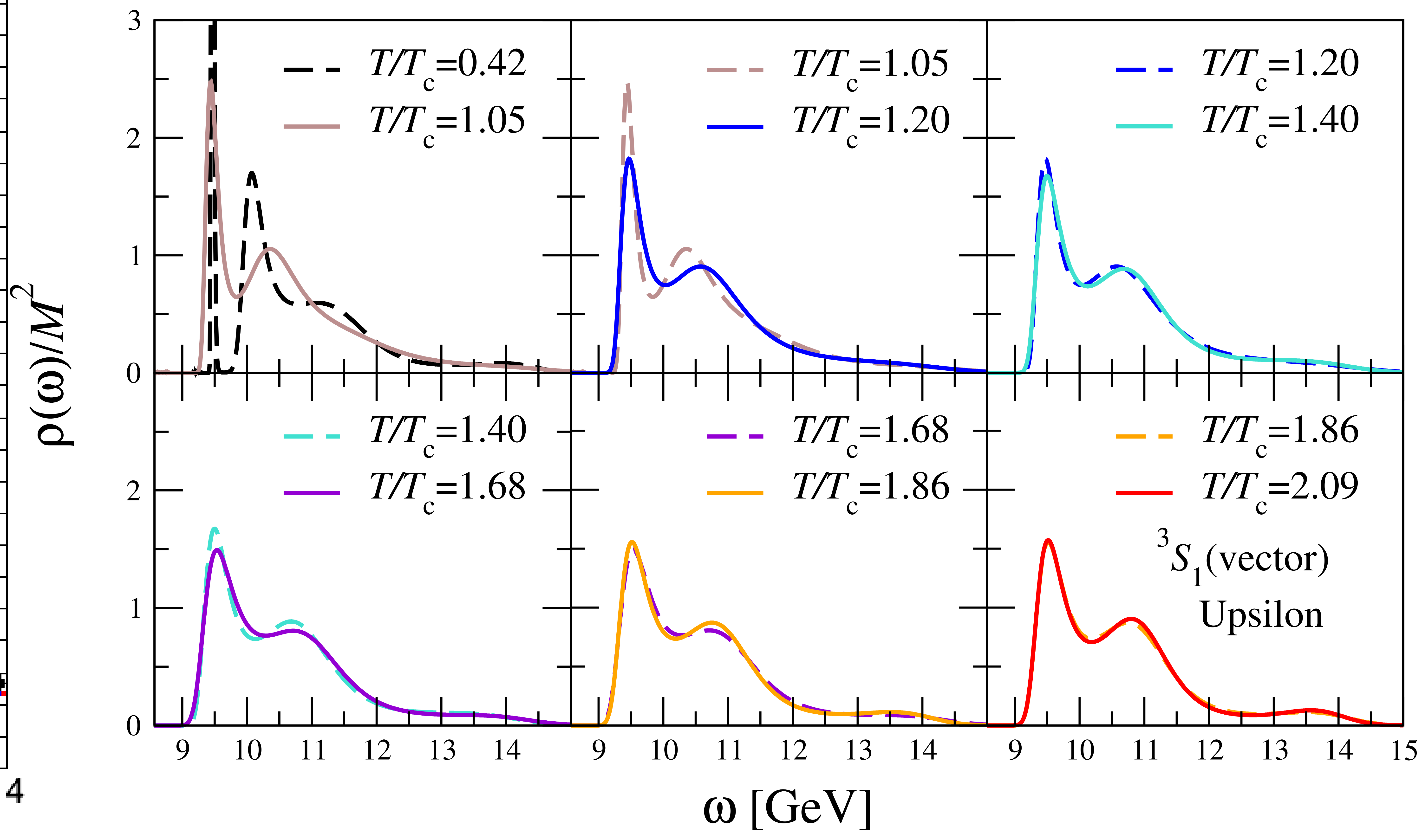
[Probing the QCD phase structure using event-by-event fluctuations] - Nayak, Tapan K. - arXiv:2008.04643

Sequential suppression of Upsilon system

CMS, PRL109 (2012) 222301



FASTSUM, JHEP07 (2014) 097 (over 100 citations)



Lattice QCD in non-zero temperature faces challenges

- Non-equilibrium QCD
- QCD in finite baryon density

Content

1. Introduction to Quantum Computing

Beginning

International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

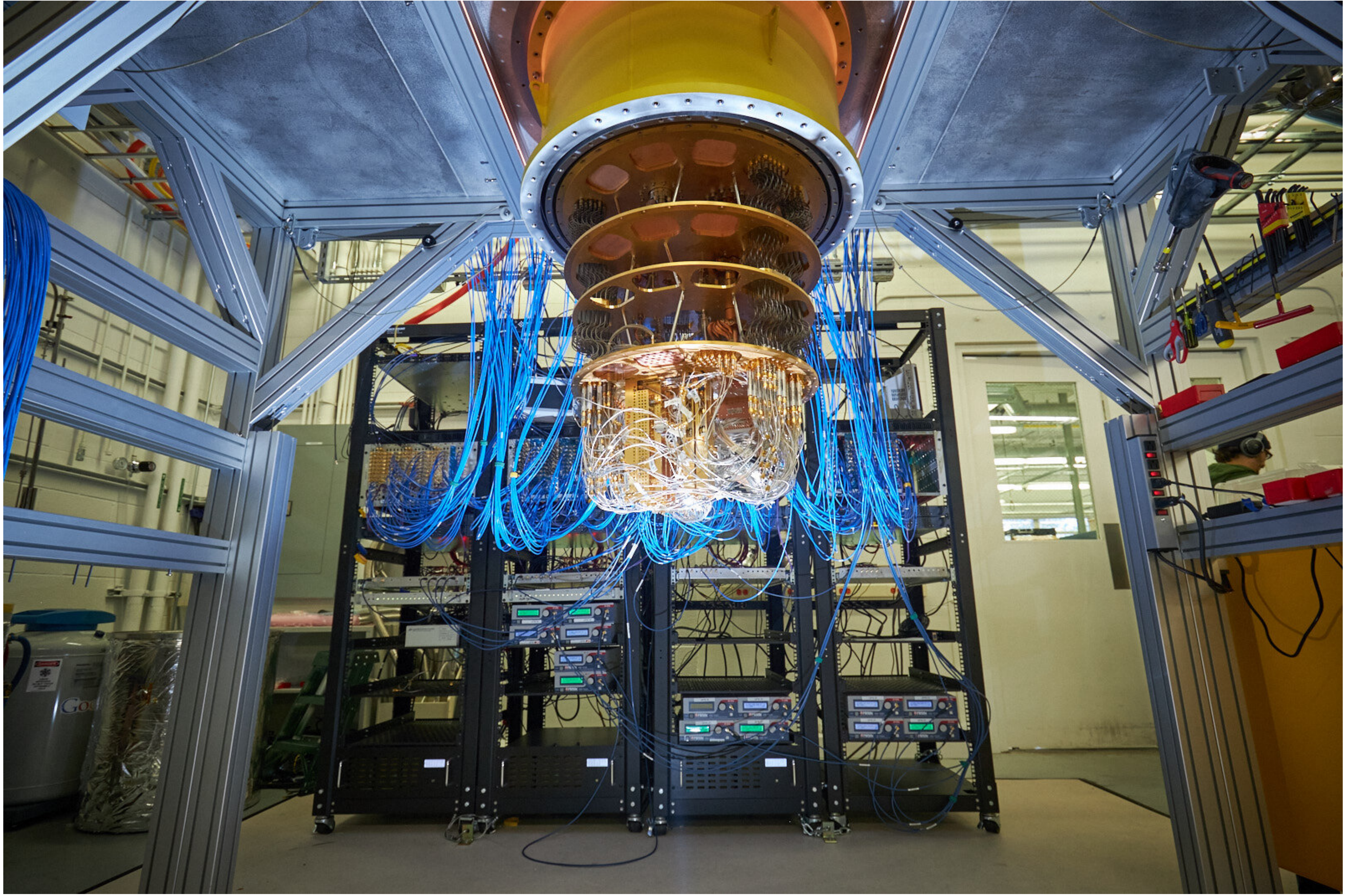
Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

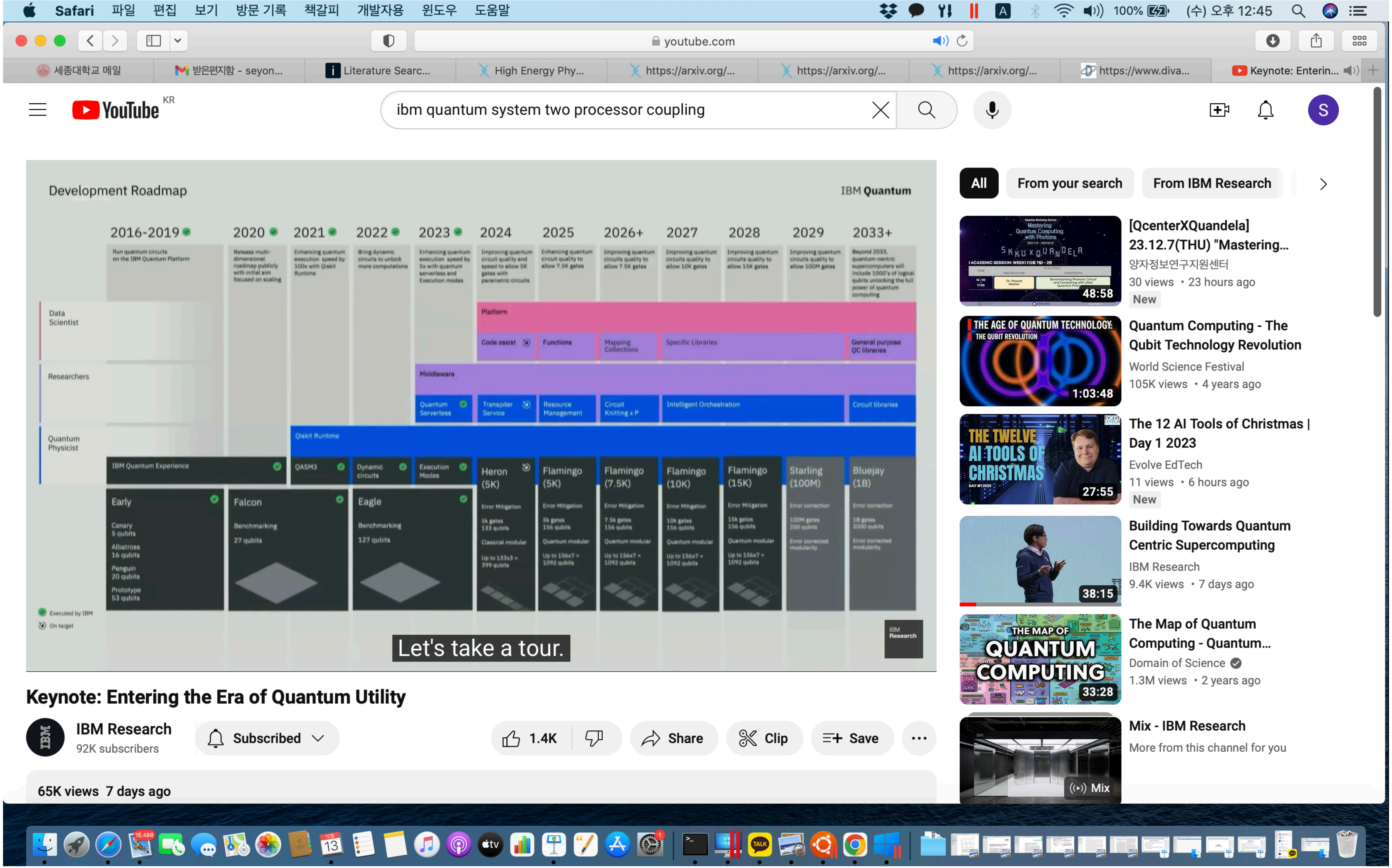
Received May 7, 1981

1. INTRODUCTION

On the program it says this is a keynote speech—and I don't know what a keynote speech is. I do not intend in any way to suggest what should be in this meeting as a keynote of the subjects or anything like that. I have my own things to say and to talk about and there's no implication that anybody needs to talk about the same thing or anything like it. So what I want to talk about is what Mike Dertouzos suggested that nobody would talk about. I want to talk about the problem of simulating physics with



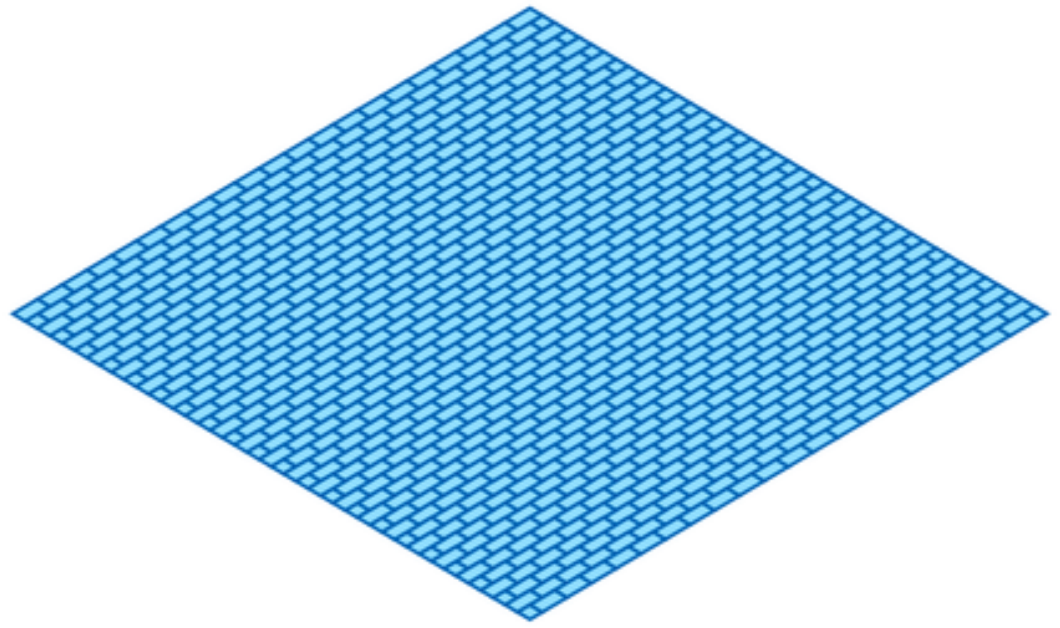
Google's Sycamore processor mounted in a cryostat, recently used to demonstrate quantum supremacy and the largest quantum chemistry simulation on a quantum computer. Credit: Rocco Ceselin



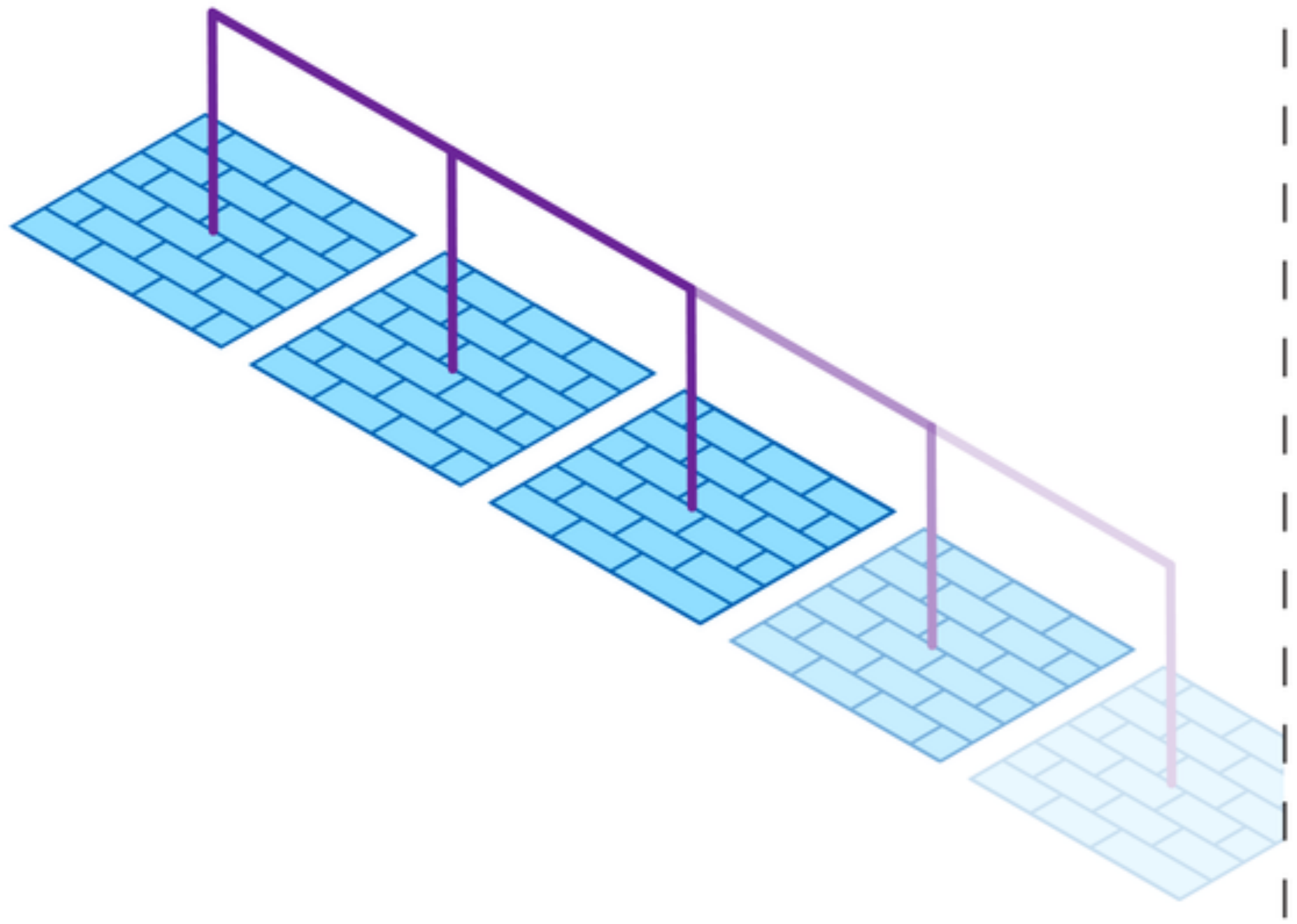
Safari 파일 편집 보기 방문 기록 책갈피 개발자용 윈도우 도움말
100% (수) 오후 12:45
youtube.com
세종대학교 메일
받은편지함 - seyon...
Literature Searc...
High Energy Phy...
https://arxiv.org/...
https://arxiv.org/...
https://arxiv.org/...
https://www.diva...
Keynote: Enterin...
YouTube KR
ibm quantum system two processor coupling
Development Roadmap
IBM Quantum
2016-2019
2020
2021
2022
2023
2024
2025
2026+
2027
2028
2029
2033+
Data Scientist
Researchers
Quantum Physicist
Platform
Code assist
Functions
Mapping Collections
Specific Libraries
General purpose QC libraries
Middleware
Quantum Serverless
Transpiler Service
Resource Management
Circuit Knitting x P
Intelligent Orchestration
Circuit libraries
Qiskit Runtime
IBM Quantum Experience
QASM3
Dynamic circuits
Execution Modes
Heron (5K)
Flamingo (5K)
Flamingo (7.5K)
Flamingo (10K)
Flamingo (15K)
Starling (100M)
Bluejay (1B)
Early
Canary 5 qubits
Albatross 16 qubits
Penguin 20 qubits
Prototype 53 qubits
Falcon
Benchmarking
27 qubits
Eagle
Benchmarking
127 qubits
Executed by IBM
On target
IBM Research
Let's take a tour.
Keynote: Entering the Era of Quantum Utility
IBM Research
92K subscribers
Subscribed
1.4K
Share
Clip
Save
65K views 7 days ago
All
From your search
From IBM Research
[QcenterXQuandela] 23.12.7(THU) "Mastering...
양자정보연구지원센터
30 views • 23 hours ago
New
48:58
Quantum Computing - The Qubit Technology Revolution
World Science Festival
105K views • 4 years ago
1:03:48
THE AGE OF QUANTUM TECHNOLOGY: THE QUBIT REVOLUTION
The 12 AI Tools of Christmas | Day 1 2023
Evolve EdTech
11 views • 6 hours ago
New
27:55
THE TWELVE AI TOOLS OF CHRISTMAS
Building Towards Quantum Centric Supercomputing
IBM Research
9.4K views • 7 days ago
38:15
The Map of Quantum Computing - Quantum...
Domain of Science
1.3M views • 2 years ago
33:28
THE MAP OF QUANTUM COMPUTING
Mix - IBM Research
More from this channel for you
Mix

2023

CONDOR
1,121 qubits

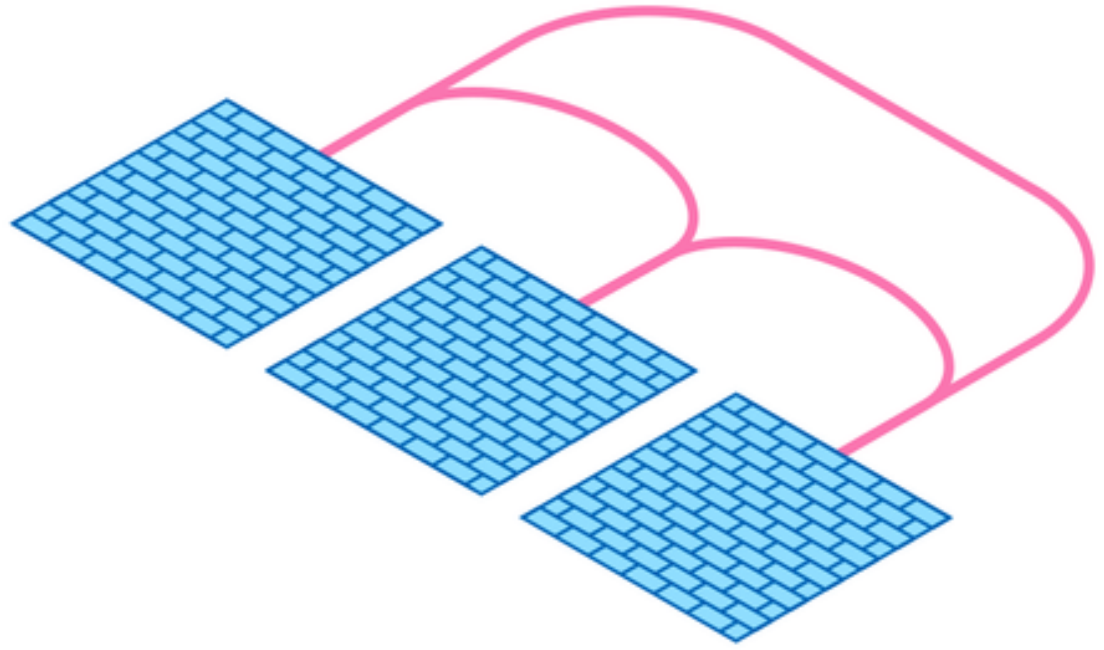


HERON
133 qubits x p

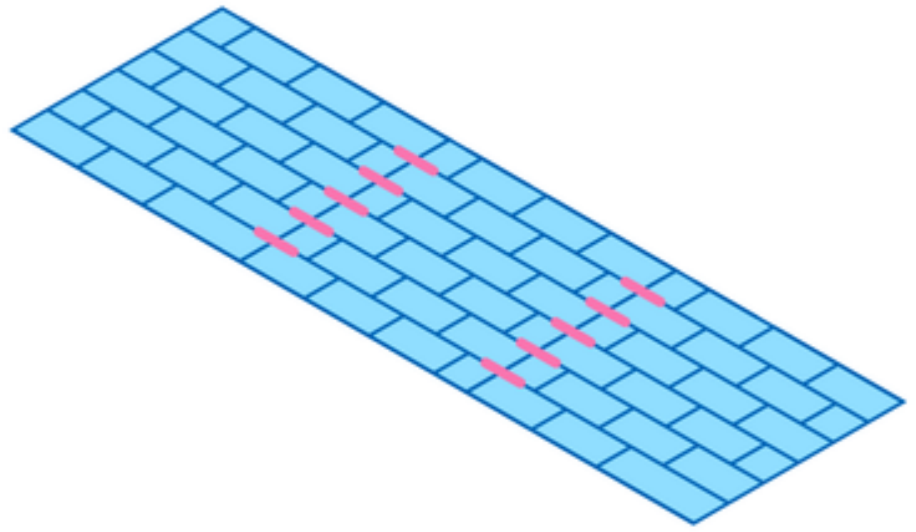


2024

FLAMINGO
1,386 qubits

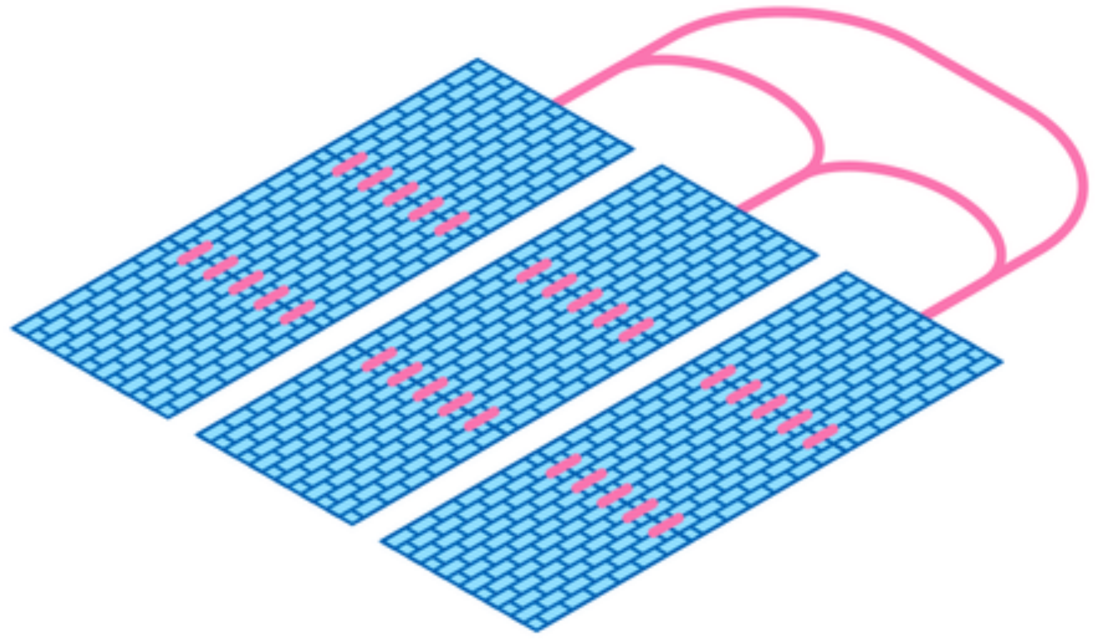


CROSSBILL
408 qubits



2025

KOOKABURRA
4,158 qubits



Communications

Classical

Qauntum

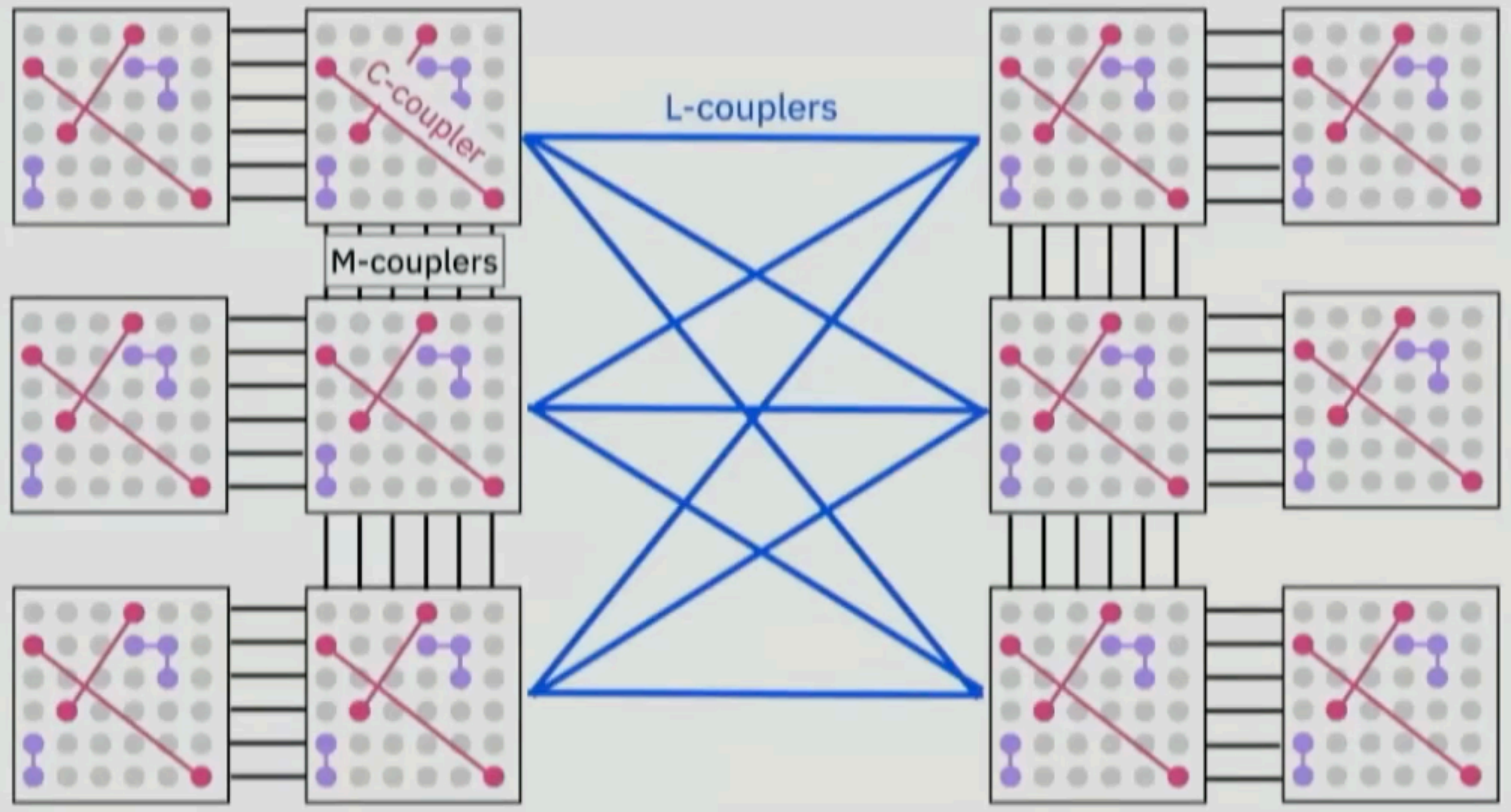
https://spectrum.ieee.org/ibm-condor

New error correction codes
and system modularity

C-coupler enables long-range on-chip connections for high-rate LDPC codes

L-coupler enables joining multiple logical memories to create large-scale systems

M-coupler enables joining multiple short-range chip-to-chip connections



IBM Quantum Summit 2023

We need a system
that has c couplers to enable long

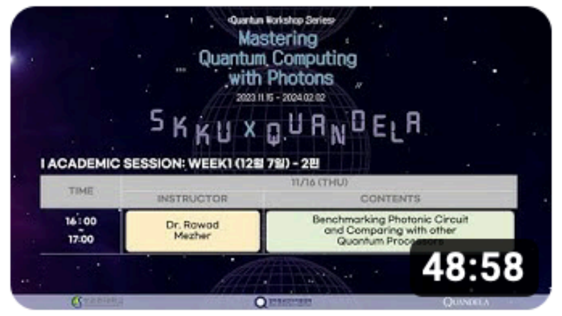
IBM Research

Keynote: Entering the Era of Quantum Utility

IBM Research
92K subscribers
Subscribed
1.4K | Share | Clip | Save

65K views 7 days ago

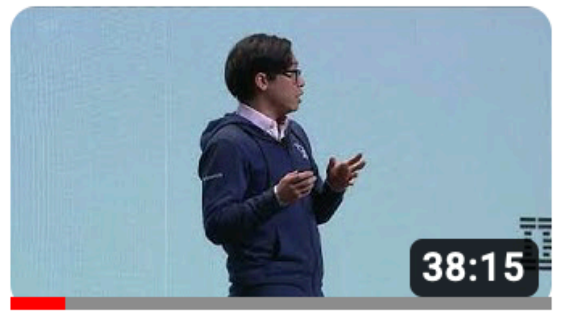
All | From your search | From IBM Research



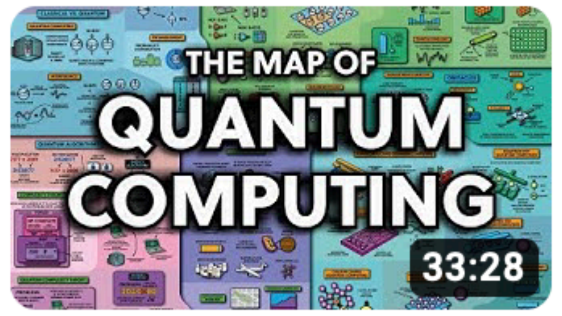
[QcenterXQuandela] 23.12.7(THU) "Mastering...
양자정보연구지원센터
30 views • 23 hours ago
New

Quantum Computing - The Qubit Technology Revolution
World Science Festival
105K views • 4 years ago

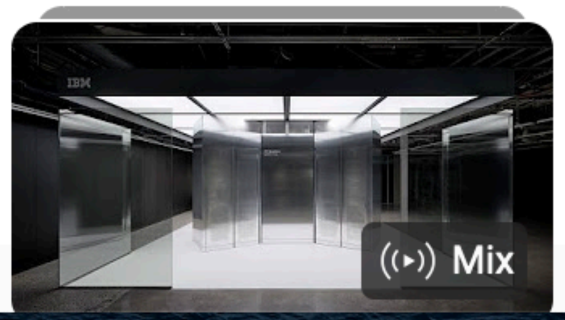
The 12 AI Tools of Christmas | Day 1 2023
Evolve EdTech
11 views • 6 hours ago
New



Building Towards Quantum Centric Supercomputing
IBM Research
9.4K views • 7 days ago



The Map of Quantum Computing - Quantum...
Domain of Science
1.3M views • 2 years ago



Mix - IBM Research
More from this channel for you

High-threshold and low-overhead fault-tolerant quantum memory

Sergey Bravyi[1], Andrew W. Cross[1], Jay M. Gambetta[1], Dmitri Maslov[1], Patrick Rall[2], and Theodore J. Yoder[1]

[1]IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 (USA)
[2]IBM Quantum, MIT-IBM Watson AI Lab, Cambridge, MA 02142 (USA)

August 16, 2023

Abstract

Quantum error correction becomes a practical possibility only if the physical error rate is below a threshold value that depends on a particular quantum code, syndrome measurement circuit, and a decoding algorithm. Here we present an end-to-end quantum error correction protocol that implements fault-tolerant memory based on a family of LDPC codes with a high encoding rate that achieves an error threshold of 0.8% for the standard circuit-based noise model. This is on par with the surface code which has remained an uncontested leader in terms of its high error threshold for nearly 20 years. The full syndrome measurement cycle for a length-n code in our family requires n ancillary qubits and a depth-7 circuit composed of nearest-neighbor CNOT gates. The required qubit connectivity is a degree-6 graph that consists of two edge-disjoint planar subgraphs. As a concrete example, we show that 12 logical qubits can be preserved for ten million syndrome cycles using 288 physical qubits in total, assuming the physical error rate of 0.1%. We argue that achieving the same level of error suppression on 12 logical qubits with the surface code would require more than 4000 physical qubits. Our findings bring demonstrations of a low-overhead fault-tolerant quantum memory within the reach of near-term quantum processors.

Logical quantum processor based on reconfigurable atom arrays

Dolev Bluvstein[1], Simon J. Evered[1], Alexandra A. Geim[1], Sophie H. Li[1], Hengyun Zhou[1,2], Tom Manovitz[1], Sepehr Ebadi[1], Madelyn Cain[1], Marcin Kalinowski[1], Dominik Hangleiter[3], J. Pablo Bonilla Ataides[1], Nishad Maskara[1], Iris Cong[1], Xun Gao[1], Pedro Sales Rodriguez[2], Thomas Karolyshyn[2], Giulia Semeghini[4], Michael J. Gullans[3], Markus Greiner[1], Vladan Vuletić[5], and Mikhail D. Lukin[1,†]

[1] *Department of Physics, Harvard University, Cambridge, MA 02138, USA*
[2] *QuEra Computing Inc., Boston, MA 02135, USA*
[3] *Joint Center for Quantum Information and Computer Science, NIST/University of Maryland, College Park, Maryland 20742, USA*
[4] *John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA*
[5] *Department of Physics and Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[†] *Corresponding Author; E-mail: lukin@physics.harvard.edu*

Suppressing errors is the central challenge for useful quantum computing [1], requiring quantum error correction [2–6] for large-scale processing. However, the overhead in the realization of error-corrected "logical" qubits, where information is encoded across many physical qubits for redundancy [2–4], poses significant challenges to large-scale logical quantum computing. Here we report the realization of a programmable quantum processor based on encoded logical qubits operating with up to 280 physical qubits. Utilizing logical-level control and a zoned architecture in reconfigurable neutral atom arrays [7], our system combines high two-qubit gate fidelities [8], arbitrary connectivity [7, 9], as well as fully programmable single-qubit rotations and mid-circuit readout [10–15]. Operating this logical processor with various types of encodings, we demonstrate improvement of a two-qubit logic gate by scaling surface code [6] distance from $d = 3$ to $d = 7$, preparation of color code qubits with break-even fidelities [5], fault-tolerant creation of logical GHZ states and feedforward entanglement teleportation, as well as operation of 40 color code qubits. Finally, using three-dimensional [[8,3,2]] code blocks [16, 17], we realize computationally complex sampling circuits [18] with up to 48 logical qubits entangled with hypercube connectivity [19] with 228 logical two-qubit gates and 48 logical CCZ gates [20]. We find that this logical encoding substantially improves algorithmic performance with error detection, outperforming physical qubit fidelities at both cross-entropy benchmarking and quantum simulations of fast scrambling [21, 22]. These results herald the advent of early error-corrected quantum computation and chart a path toward large-scale logical processors.

Example of QC : Shor's Algorithm

- RSA algorithm —> integer factoring problem
- change integer factoring problem into order-finding problem using a digital computer
- solve order-finding problem using a Quantum Computer
- P. Shor, proceedings of 35th annual symposium on the foundations of computer science, 1994, p124-134

How to factor $N = pq$, a product of two primes

https://www.youtube.com/watch?v=-UrdExQW0cs&t=712s

1. Make a guess, $g < N$ that shares no factors with N
2. Find r such that $g^r = mN + 1$
3. If r is even, calculate $(g^{r/2} + 1)$ and $(g^{r/2} - 1)$. If r is odd, go back to step 1
4. Use Euclid's algorithm to find the greatest common divisor of $g^{r/2} + 1$ and N

How to factor N, a product of two primes (example)

https://www.youtube.com/watch?v=-UrdExQW0cs&t=712s

$N = 77 = pq$

1. $g = 8$
2. Find r such that $g^r = mN + 1$ —〉 $r = 10$
3. $(g^{r/2} + 1) = 32{,}769$, $(g^{r/2} - 1) = 32{,}767$, $(g^{r/2} + 1)(g^{r/2} - 1) = mN$
4. Use Euclid's algorithm to find the greatest common divisor of $g^{r/2} + 1$ and N:

   $32769/77 = 425R44 -> 77/44 = 1R33 -> 44/33 = 1R11 -> 33/11 = 3R0$
5. 11 is a prime factor of 77!

How to factor N, a product of two primes (example)

https://www.youtube.com/watch?v=-UrdExQW0cs&t=712s

- QC is better at step 3, finding r of

  $g^r = mN + 1$, because

  $g, g^1, g^2, \cdots, g^{r-1}, g^r, g^{r+1}, \cdots, g^{2r}, \cdots, g^{3r}, \cdots,$

  are periodic
- Finding r is related to Fourier transform and QC Fourier transform is faster
- Fast Fourier transform in DC and quantum Fourier transform in QC!

For digital computer,

the best Fourier transform algorithm is Cooley−Tukey algorithm and the number of operations is

$$\sim N \times \log_2 N, \quad N = 2^n$$

Cooley–Tukey Algorithm

$ex) x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, \quad N = 2^3, m = 3$

For $k = 1, \cdots, N,$

$$X_k = \sum_{n=0}^{n=7} x_n (\omega_N)^{-kn}, \ \ \omega_N = e^{\frac{2\pi i}{N}} = e^{\frac{2\pi i}{8}} = e^{\frac{i\pi}{4}}$$

$$= \sum_{m=0}^{m=3} x_{2m} \omega_N^{-k(2m)} + \omega_N^{-k} \sum_{m=0}^{m=3} x_{2m+1} \omega_N^{-k(2m)} = X_k^e + \omega_N^{-k} \ X_k^o$$

Note that $X_{k+N/2} = X_k^e + \omega_N^{-k} \ X_k^o$

$$X_k^e = X_k^{ee} + X_k^{eo} \ \omega_{\frac{N}{2}}^{-k} = X_k^{ee} + X_k^{eo} \ \omega_N^{-2k} \text{ since } \omega_{\frac{N}{2}} = e^{\frac{2\pi i}{\frac{N}{2}}} = e^{\frac{4\pi i}{N}} = \omega_N^2$$

Similarly, $X_k^o = X_k^{oe} + X_k^{oo} \ \omega_{\frac{N}{2}}^{-k} = X_k^{oe} + X_k^{oo} \ \omega_N^{-2k}$

Cooley−Tukey Algorithm

$ex) x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, \quad N = 2^3, m = 3,$

For $k = 1, \cdots, N,$

$$X_k = x_0\omega_N^0 + x_1\omega_N^{-k} + x_2\omega_N^{-2k} \cdots + x_7\omega_N^{-7k}$$

$$= X_k^e + X_k^o\ \omega_N^{-k}$$

$$= (X_k^{ee} + X_k^{eo}\ \omega_N^{-2k}) + \omega_N^{-k}(X_k^{oe} + X_k^{oo}\ \omega_N^{-2k})$$

Note that since

$$X_k^{ee} = x_0 + x_4\omega_N^{-4k}, X_k^{eo} = x_2 + x_6\omega_N^{-4k}, X_k^{oe} = x_1 + x_5\omega_N^{-4k}, X_k^{oo} = x_3 + x_7\omega_N^{-4k}$$

$$X_k = (x_0 + x_4\omega_N^{-4k}) + \omega_N^{-2k}(x_2 + x_6\omega_N^{-4k}) + \omega_N^{-k}[(x_1 + x_5\omega_N^{-4k}) + \omega_N^{-2k}(x_3 + x_7\omega_N^{-4k})]$$

$$= x_0 + x_4\omega_N^{-4k} + x_2\omega_N^{-2k} + x_6\omega_N^{-6k} + x_1\omega_N^{-k} + x_5\omega_N^{-5k} + x_3\omega_N^{-3k} + x_7\omega_N^{-7k}$$

Cooley−Tukey Algorithm

Bit−ordering —〉 bit−reverse, swap so that the data x_i ordered as bit−reversed

$n = 0, (0,0,0) \rightarrow (0,0,0), n = 0$

$n = 1, (0,0,1) \rightarrow (1,0,0), n = 4$

$n = 2, (0,1,0) \rightarrow (0,1,0), n = 2$

$n = 3, (0,1,1) \rightarrow (1,1,0), n = 6$

$n = 4, (1,0,0) \rightarrow (0,0,1), n = 1$

$n = 5, (1,0,1) \rightarrow (1,0,1), n = 5$

$n = 6, (1,1,0) \rightarrow (0,1,1), n = 3$

$n = 7, (1,1,1) \rightarrow (1,1,1), n = 7$

$x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7 \rightarrow x_0, x_4, x_2, x_6, x_1, x_5, x_3, x_7$

Cooley–Tukey Algorithm

1. Bit-ordering —> bit-reverse, swap so that the data x_i ordered as bit-reversed
2. Then, form the first loop

   $(x_0 + x_4\omega_N^{-4k}), (x_2 + x_6\omega_N^{-4k}), (x_1 + x_5\omega_N^{-4k}), (x_3 + x_7\omega_N^{-4k})$

3. Then, form the 2nd loop

$(x_0 + x_4\omega_N^{-4k}) + \omega_N^{-2k}(x_2 + x_6\omega_N^{-4k}),\ (x_1 + x_5\omega_N^{-4k}) + \omega_N^{-2k}(x_3 + x_7\omega_N^{-4k})$

4. Then, form the 3rd loop

$[(x_0 + x_4\omega_N^{-4k}) + \omega_N^{-2k}(x_2 + x_6\omega_N^{-4k})] + \omega_N^{-k}[(x_1 + x_5\omega_N^{-4k}) + \omega_N^{-2k}(x_3 + x_7\omega_N^{-4k})]$

Cooley–Tukey Algorithm

$$[(x_0 + x_4\omega_N^{-4k}) + \omega_N^{-2k}(x_2 + x_6\omega_N^{-4k})] + \omega_N^{-k}[(x_1 + x_5\omega_N^{-4k}) + \omega_N^{-2k}(x_3 + x_7\omega_N^{-4k})]$$

$$k = 0,\ [(x_0 + x_4) + (x_2 + x_6)] + [(x_1 + x_5) + (x_3 + x_7)]$$

$$k = 1,\ [(x_0 + x_4\omega_N^{-4}) + \omega_N^{-2}(x_2 + x_6\omega_N^{-4})] + \omega_N^{-1}[(x_1 + x_5\omega_N^{-4}) + \omega_N^{-2}(x_3 + x_7\omega_N^{-4})]$$

$$k = 2,\ [(x_0 + x_4\omega_N^{-8}) + \omega_N^{-4}(x_2 + x_6\omega_N^{-8})] + \omega_N^{-2}[(x_1 + x_5\omega_N^{-8}) + \omega_N^{-4}(x_3 + x_7\omega_N^{-8})]$$

$$k = 3,\ [(x_0 + x_4\omega_N^{-12}) + \omega_N^{-6}(x_2 + x_6\omega_N^{-12})] + \omega_N^{-3}[(x_1 + x_5\omega_N^{-12}) + \omega_N^{-6}(x_3 + x_7\omega_N^{-12})]$$

$$k = 4,\ [(x_0 + x_4\omega_N^{-16}) + \omega_N^{-8}(x_2 + x_6\omega_N^{-16})] + \omega_N^{-4}[(x_1 + x_5\omega_N^{-16}) + \omega_N^{-8}(x_3 + x_7\omega_N^{-16})]$$

$$k = 5,\ [(x_0 + x_4\omega_N^{-20}) + \omega_N^{-10}(x_2 + x_6\omega_N^{-20})] + \omega_N^{-5}[(x_1 + x_5\omega_N^{-20}) + \omega_N^{-10}(x_3 + x_7\omega_N^{-20})]$$

$$k = 6,\ [(x_0 + x_4\omega_N^{-24}) + \omega_N^{-12}(x_2 + x_6\omega_N^{-24})] + \omega_N^{-6}[(x_1 + x_5\omega_N^{-24}) + \omega_N^{-12}(x_3 + x_7\omega_N^{-24})]$$

$$k = 7,\ [(x_0 + x_4\omega_N^{-28}) + \omega_N^{-14}(x_2 + x_6\omega_N^{-28})] + \omega_N^{-7}[(x_1 + x_5\omega_N^{-28}) + \omega_N^{-14}(x_3 + x_7\omega_N^{-28})]$$

Quantum Fourier transform

$$[(x_0 + x_4\omega_N^{-4k}) + \omega_N^{-2k}(x_2 + x_6\omega_N^{-4k})] + \omega_N^{-k}[(x_1 + x_5\omega_N^{-4k}) + \omega^{-2k}(x_3 + x_7\omega_N^{-4k})],\ N = 2^3 = 8$$

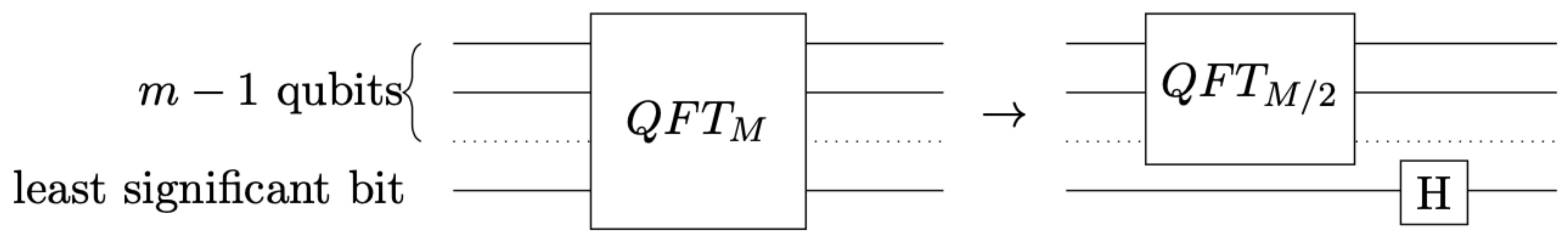


Figure 5.4: $QFT_{M/2}$ and a Hadamard gate correspond to $FFT_M/2$ on the odd and even terms

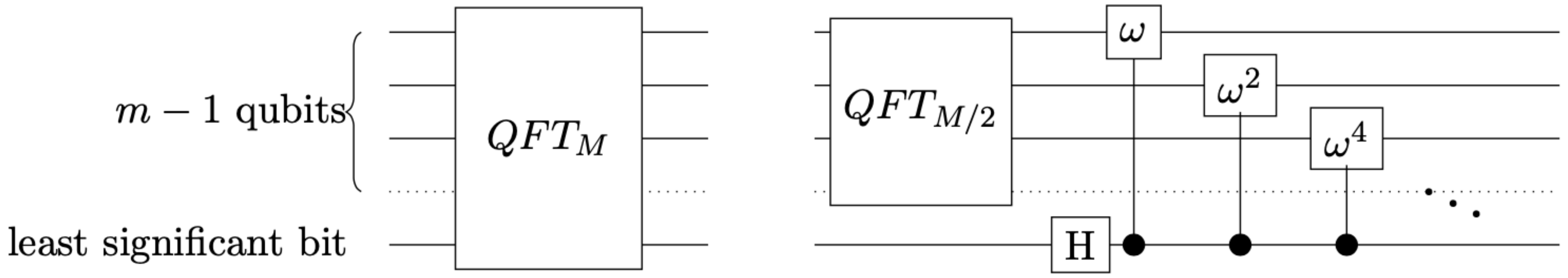


Figure 5.5: QFT_M is reduced to $QFT_{M/2}$ and M additional gates

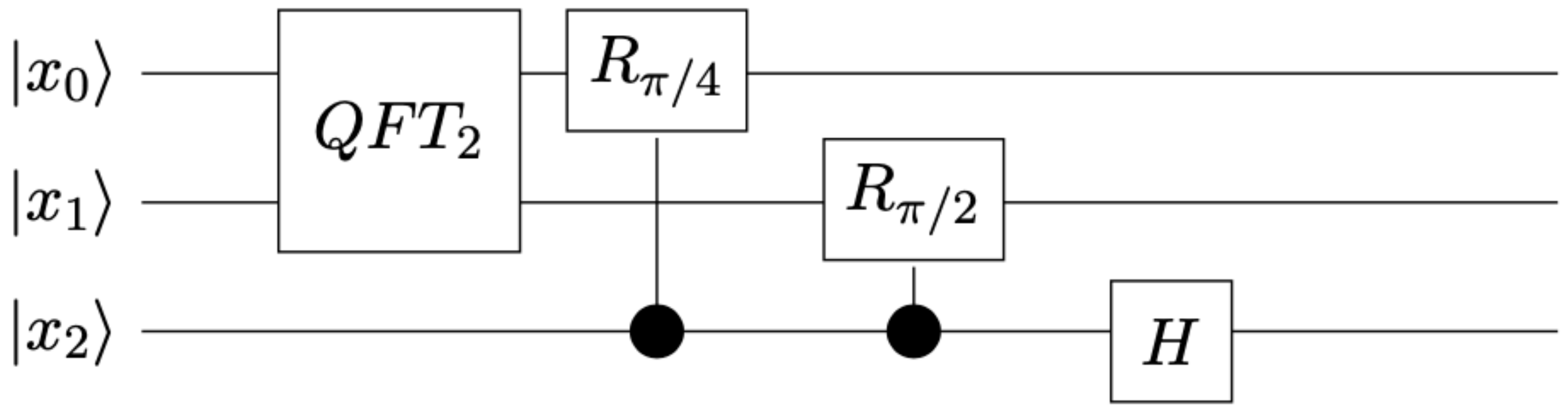


Figure 5.6: First Iteration

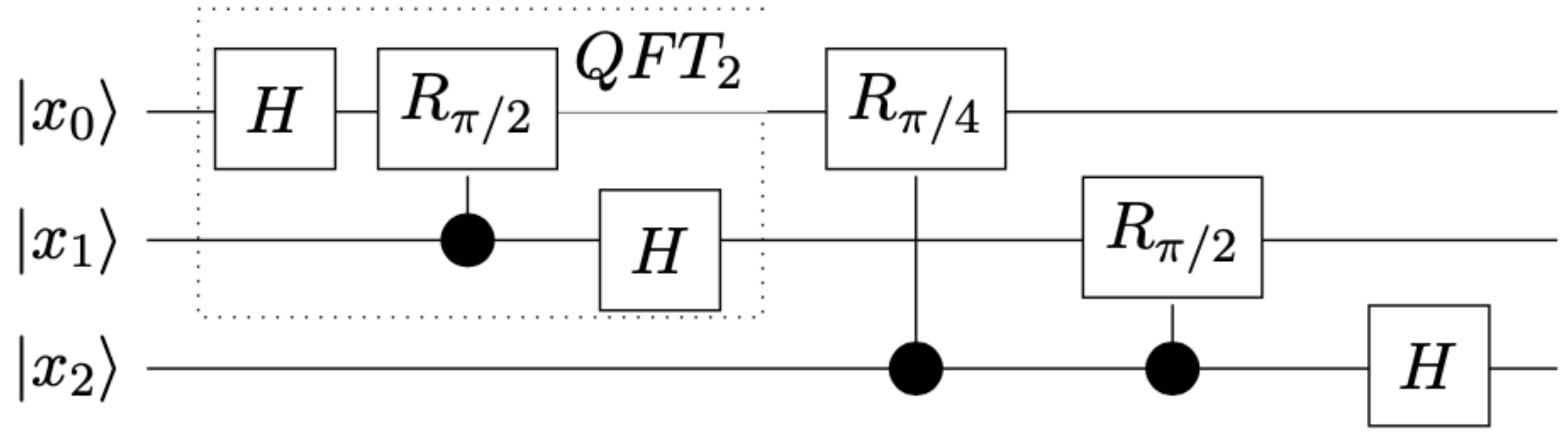
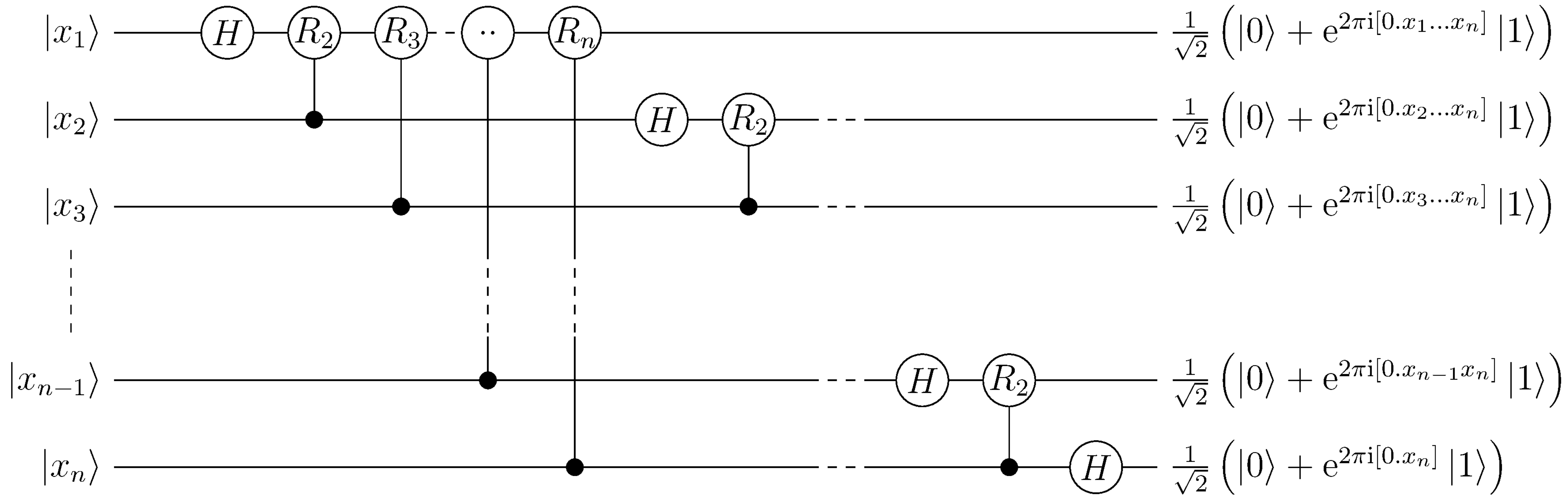


Figure 5.7: Second Iteration. Recall that $H = QFT_1$

https://courses.edx.org/c4x/BerkeleyX/CS191x/asset/chap5.pdf



https://en.wikipedia.org/wiki/Quantum_Fourier_transform

QFT for Shor's Algorithm

$$|0>|0> \;\rightarrow\; \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x>|0> \text{ under } QFT_N$$

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x>|0> \;\rightarrow\; \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x>|f(x)> \text{ under } U_f$$

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x>|f(x)> \;\rightarrow\; \sqrt{\frac{r}{N}}\sum_{i=0}^{N/r-1}|ir+x_0>|f(x_0)>$$

under measurement on $|f(x)>$

Example: QFT for Shor's algorithm

$$|0>|0> \;\; \rightarrow \;\; \frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x>|0> \text{ under } QFT_8$$

$$\frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x>|0> \;\; \rightarrow \;\; \frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x>|f(x)> \text{ under } U_f\,, f(x) = x(mod\ 2)$$

Note that $x(mod\ 2) = 0$ or 1. Choose 1

$$\frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x>|1> \;\; \rightarrow \;\; \frac{1}{2}(|1>+|3>+|5>+|7>)|1> \text{ under measurement on } f$$

$$\frac{1}{2}(|1>+|3>+|5>+|7>) \;\; \rightarrow \;\; |0>-|4> \text{ under } QFT_8$$

Example: QFT for Shor's algorithm

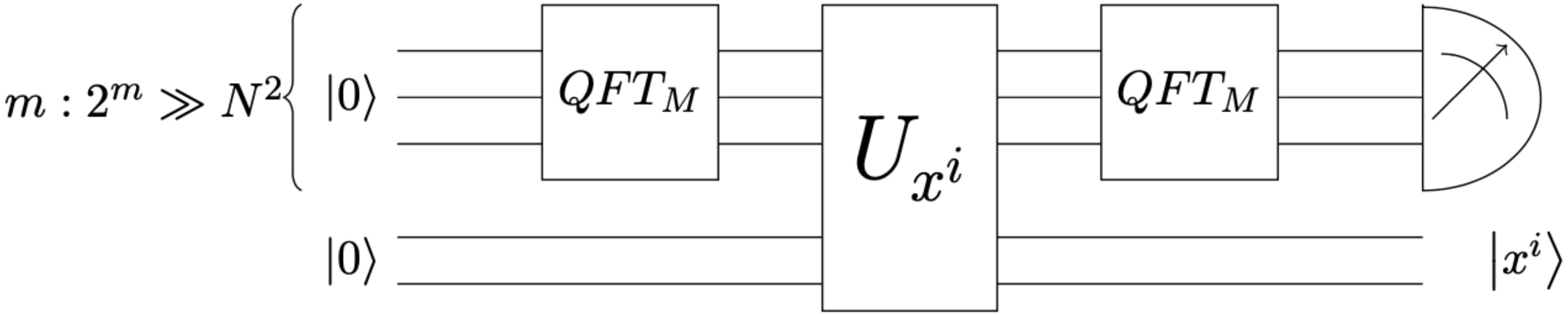


Figure 5.9: Circuit for factoring

Fourier transform of the same signal

requires

$\sim 2^n \times \log_2 2^n$ number of operations

for digital computer

$\sim n^2$ number of operations

for quantum computer

2. Quantum Error Correction, Threshold Probability and Statistical Mechanics Model

Quantum Computing in "noisy environment" or Fault–Tolerant QC

- Fighting quantum decoherence with entanglement
- Quantum Error Correction (QEC)

cf. B.M Terhal, Rev. Mod. Phys. 87 (2015) 307

QEC: Shor's code (classical counter-part)

- Smallest classical code, $|0>_L = |000>$, $|1>_L = |111>$
- Measure each bit and do majority choice
- Not applicable to quantum computer
- cf. P. W. Shor, "Fault-tolerant quantum computation" in the proceedings of 37th FOCS, p55-65

QEC: Shor's code (9−bit concatenated code)

- $|0>_L = |000>$, $|1>_L = |111>$ (entangled qubits)
- Not measure data qubits, but measure parity of data without collapsing qubits supperposition/entanglement, $|\psi>_L = \cos\frac{\theta}{2}|0>_L + \sin\frac{\theta}{2}e^{\phi}|1>_L$
- Measuring any of qubits by Z_1, Z_2, Z_3 collapses entangled data qubits
- Measuring the ancilla qubit collapses entangled data qubits if 1 (or 2, or 3) data qubit is entangled with the ancilla qubit (i.e. measuring Z_1, or (Z_2, Z_3))
- Measuring ancilla qubit doesn't collapse entangled data qubits if 1,2 (or 2, 3) data qubits are entangled with the ancilla qubit (i.e. measuring Z_1Z_2, or Z_2Z_3)

$Z_1 Z_2$ parity check

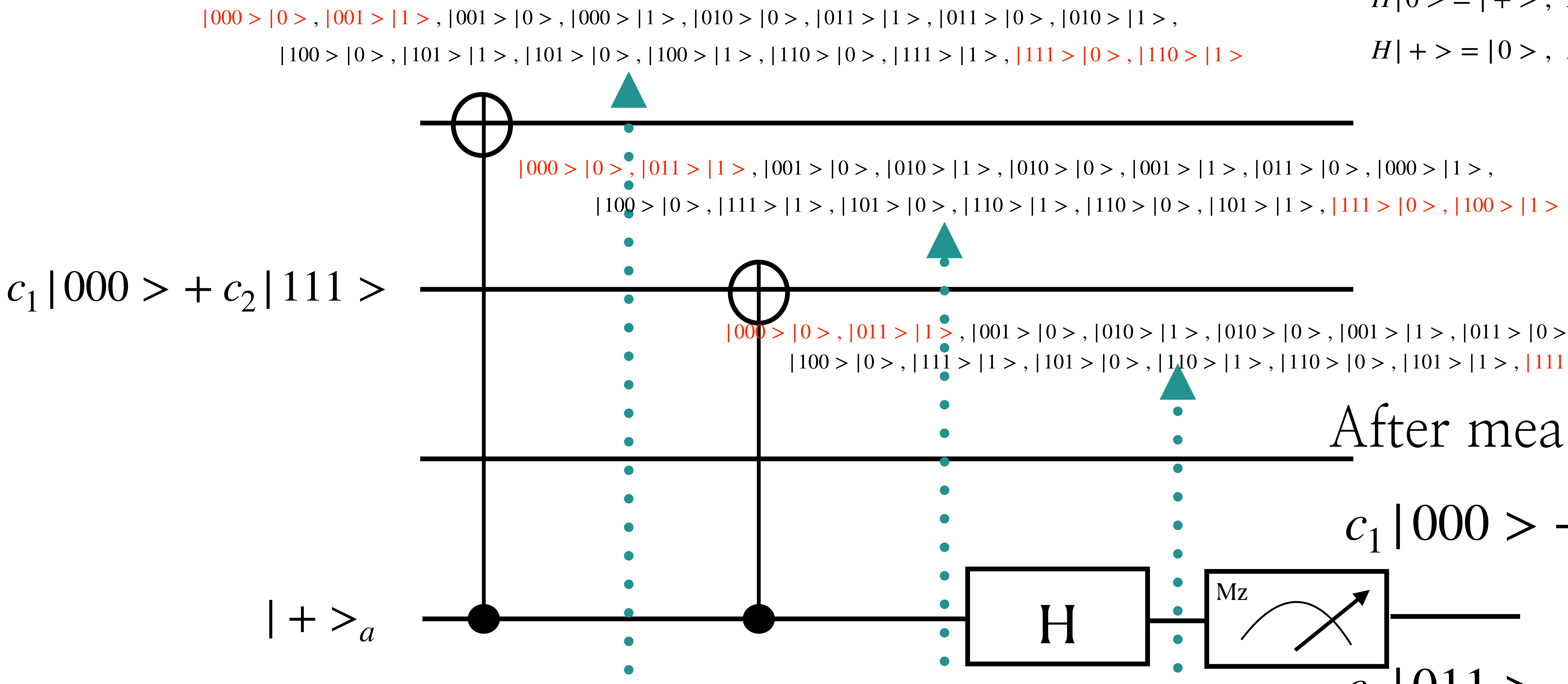
X_1X_2 parity check

Before measurement

$$|+> = \frac{1}{\sqrt{2}}(|0> + |1>),\ \ |-> = \frac{1}{\sqrt{2}}(|0> - |1>)$$

$$H|0> = |+>,\ \ H|1> = |->$$

$$H|+> = |0>,\ \ H|-> = |1>$$



After measurement

$$c_1|000> + c_2|111>$$

$$c_1|011> + c_2|100>$$

Quantum error and statistical model

- Specific quantum code with stabilizer formalism
- Modeling quantum error pattern
- Mapping quantum error pattern to statistical model
- cf. simple case: Dennis et al, J. Math. Phys. 43 (2002) 4452

Quantum Error Detection/Correction

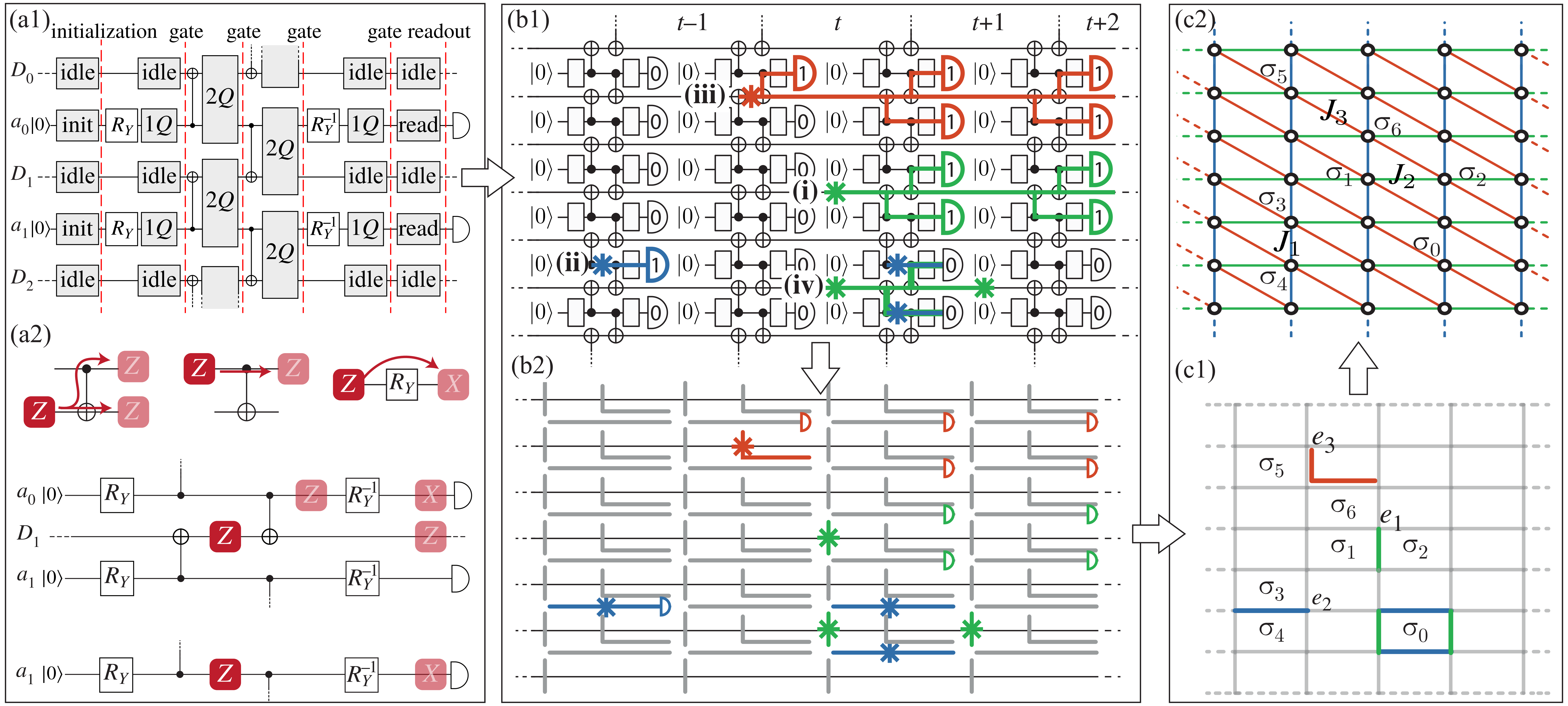
- Check whether error happens via the measurement of "ancilla" qubits: measurement result is called syndrome (quantum error detection)
- From the syndrome, guess quantum error probabilistically
- Correct quantum error

Error rate and threshold probability

- If the quantum error rate is higher than the "threshold probability", QEC is not possible.
- Above the threshold probability, "probabilistic correction" is not possible.
- "Probabilistic interpretation" is related to some statistical model

M. Rispler, D. Vodola, SK, M. Muller, "Fundamental Thresholds of Realistic Quantum Error Correction Circuits from Classical Spin Models", Quantum 6 (2022), 618

- realistic quantum circuit diagram for 1-D repetition code with phase flip error and mapping to a statistical model (quenched 2-D Ising model on a triangular lattice)



1-d repetition code and correlated error

- Protecting against phase flip error (Z-error)
- Realistic quantum circuit which implements the algorithm
- Analysis of the correlated quantum error from 1-qubit error, 1-qubit gate error, 2-qubit gate error and etc
- Mapping into 2-d random bond Ising model on triangular lattice

cf. J. Kelly et al (Google Quantum AI), "State preservation by repetitive error detection in a superconducting quantum circuit", Nature 519 (2015) 66

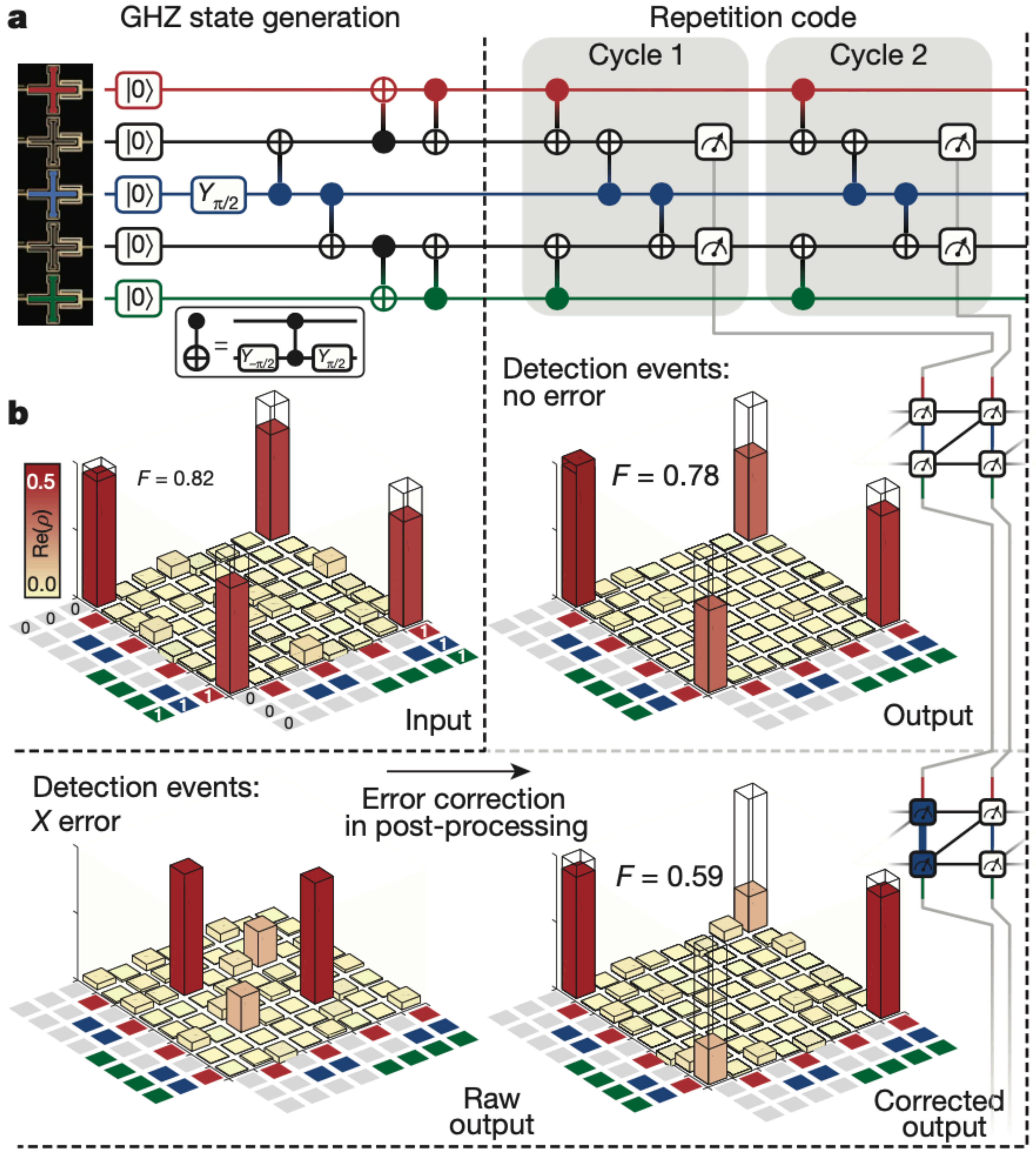


Figure 3 | Protecting the GHZ state from bit-flip errors. **a,** Quantum circuit for generating the GHZ state and two cycles of the repetition code. CNOT gates are physically implemented with controlled-phase (CZ) and single qubit gates. **b,** Quantum state tomography on the input (top left 'Input', left of black dashed line), and after the repetition code conditional on the detection events (between black dashed lines): we input a GHZ state with a fidelity (F) of 82%, and find, for the case of no detection events (top right 'Output', above grey dashed line), a 78% fidelity GHZ state. For the detection event connecting both measurement qubits (bottom left 'Raw output', below grey dashed line), indicating a likely bit-flip error on the central data qubit, we find that through cf. J. Kelly et al (Google Quantum AI), “Exponential suppression of bit or phase errors with cyclic error correction ”, Nature 595 (2021) 383

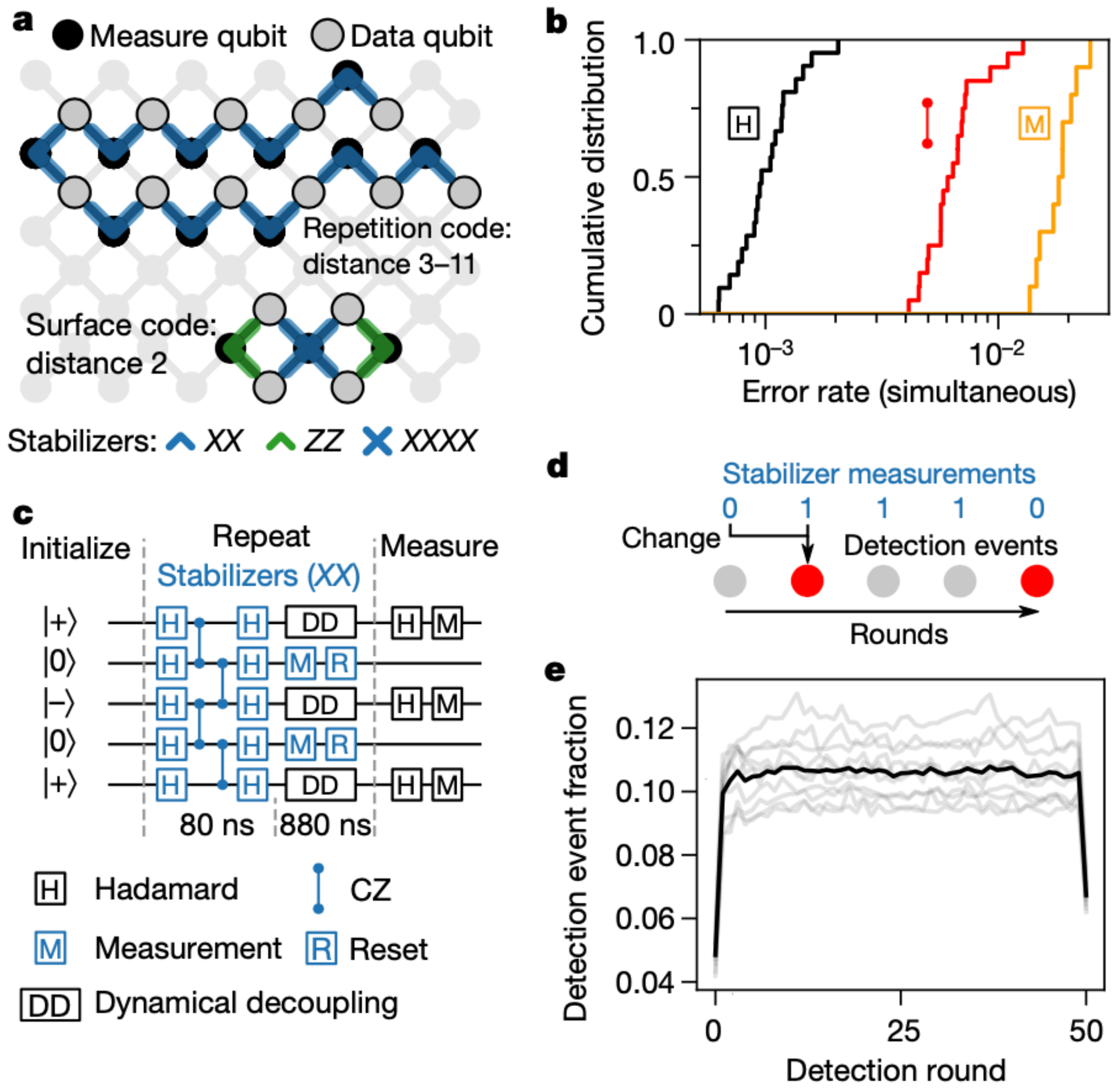
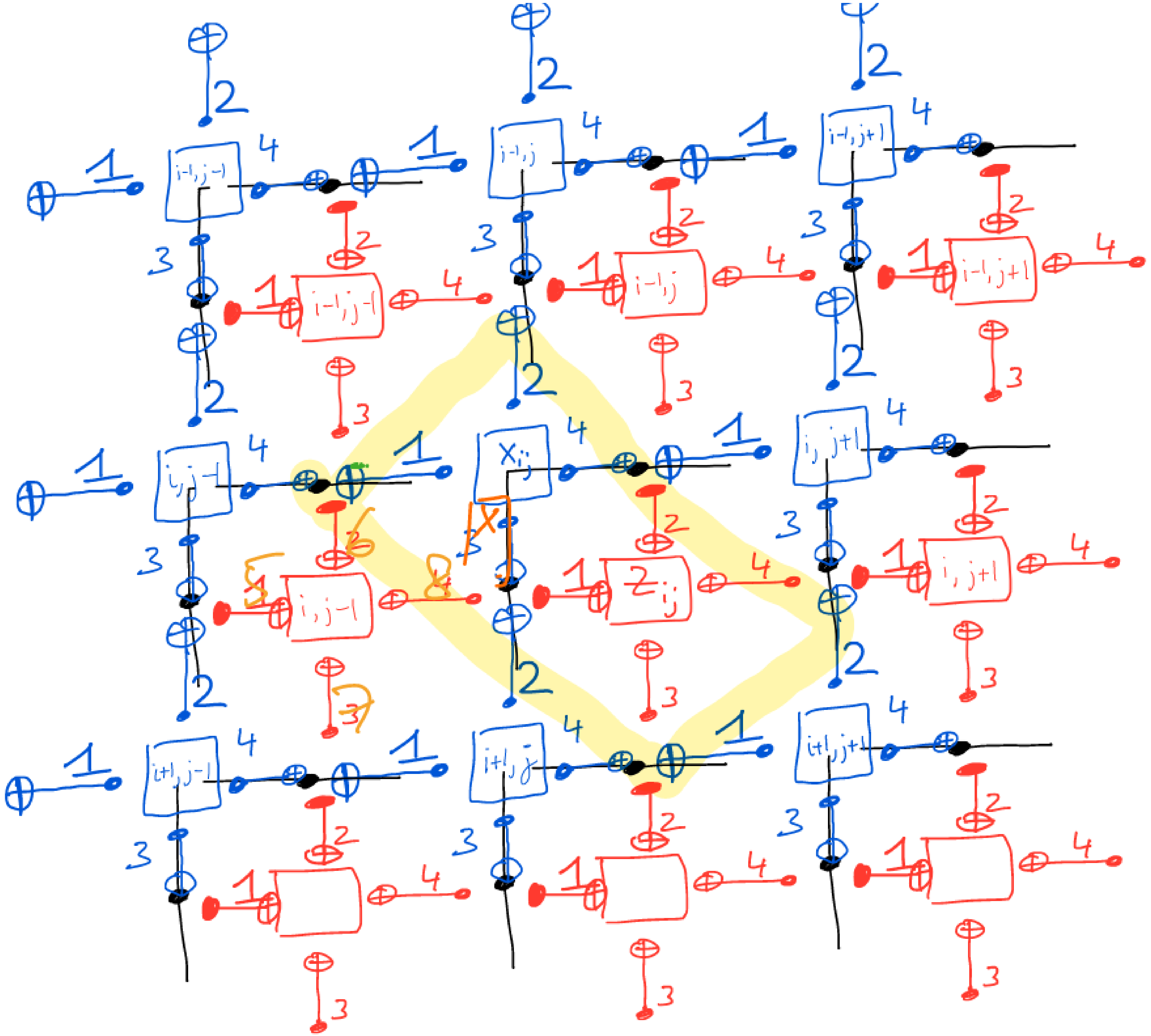


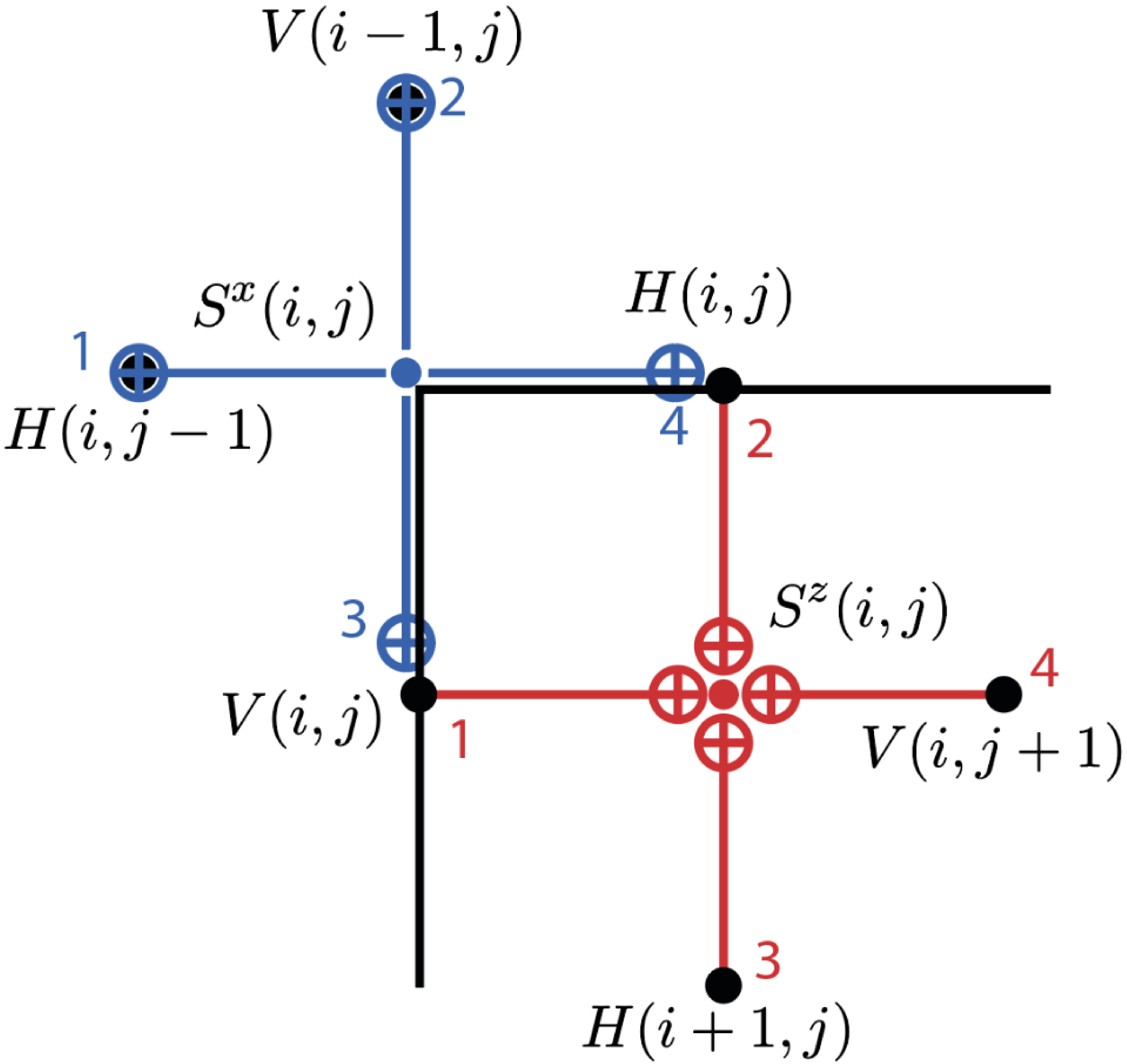
Fig. 1 | Stabilizer circuits on Sycamore. a, Layout of distance-11 repetition code and distance-2 surface code in the Sycamore processor. In the experiment, the two codes use overlapping sets of qubits, which are offset in the figure for clarity. **b**, Pauli error rates for single-qubit and CZ gates and identification error rates for measurement, each benchmarked in simultaneous operation. **c**, Circuit schematic for the phase-flip code. Data qubits are randomly initialized into $|+\rangle$ or $|-\rangle$, followed by repeated application of *XX* stabilizer measurements and finally *X*-basis measurements of the data qubits. Hadamard refers to the Hadamard gate, a quantum operation. **d**, Illustration of error detection events that occur when a measurement disagrees with the previous round. **e**, Fraction of measurements (out of 80,000) that detected an error versus measurement round for the $d=11$

3. Surface code/Toric code and Quenched $Z(2) \times Z(2)$ gauge theory

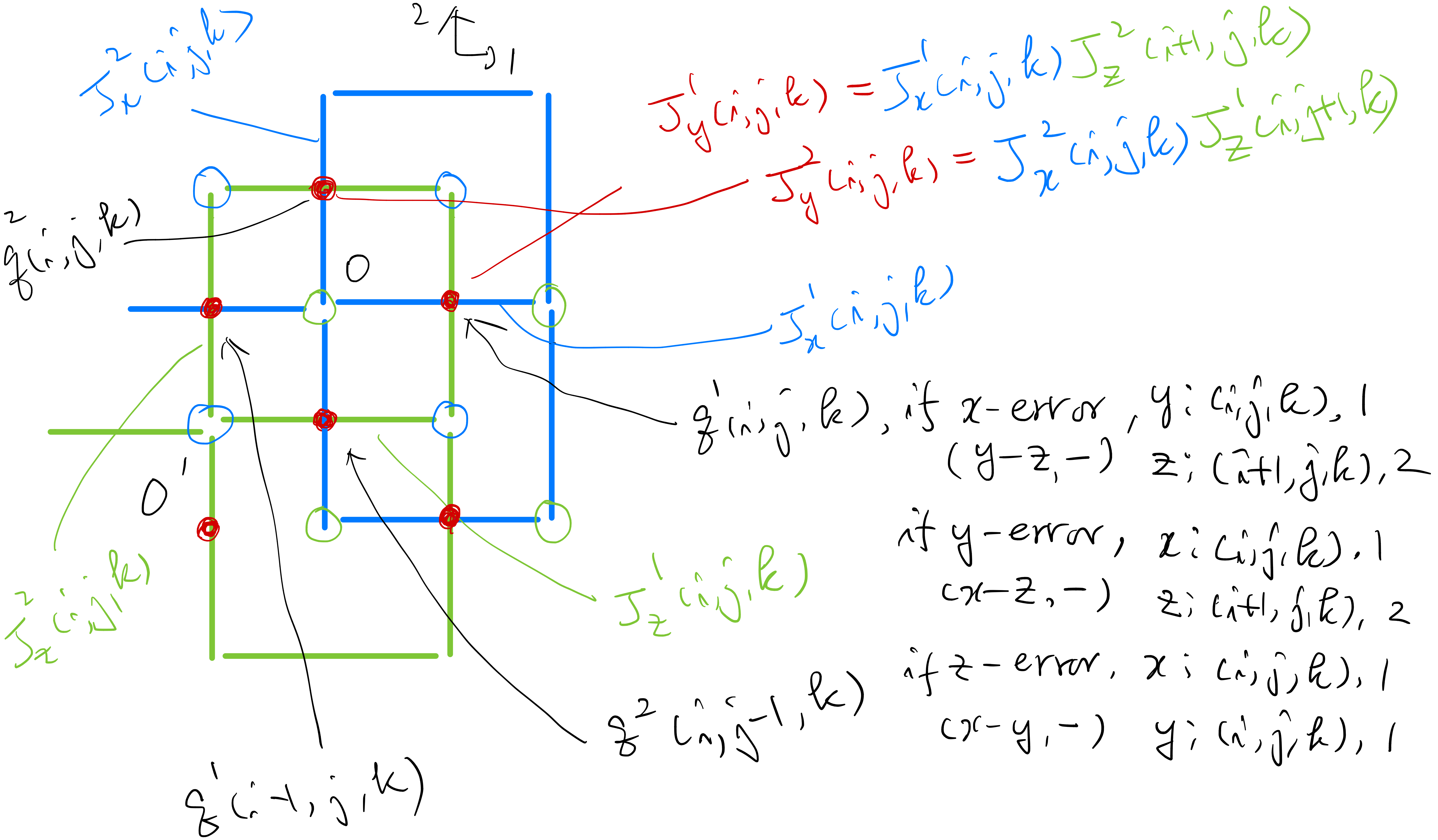
Surface code(Toric Code) circuit layout



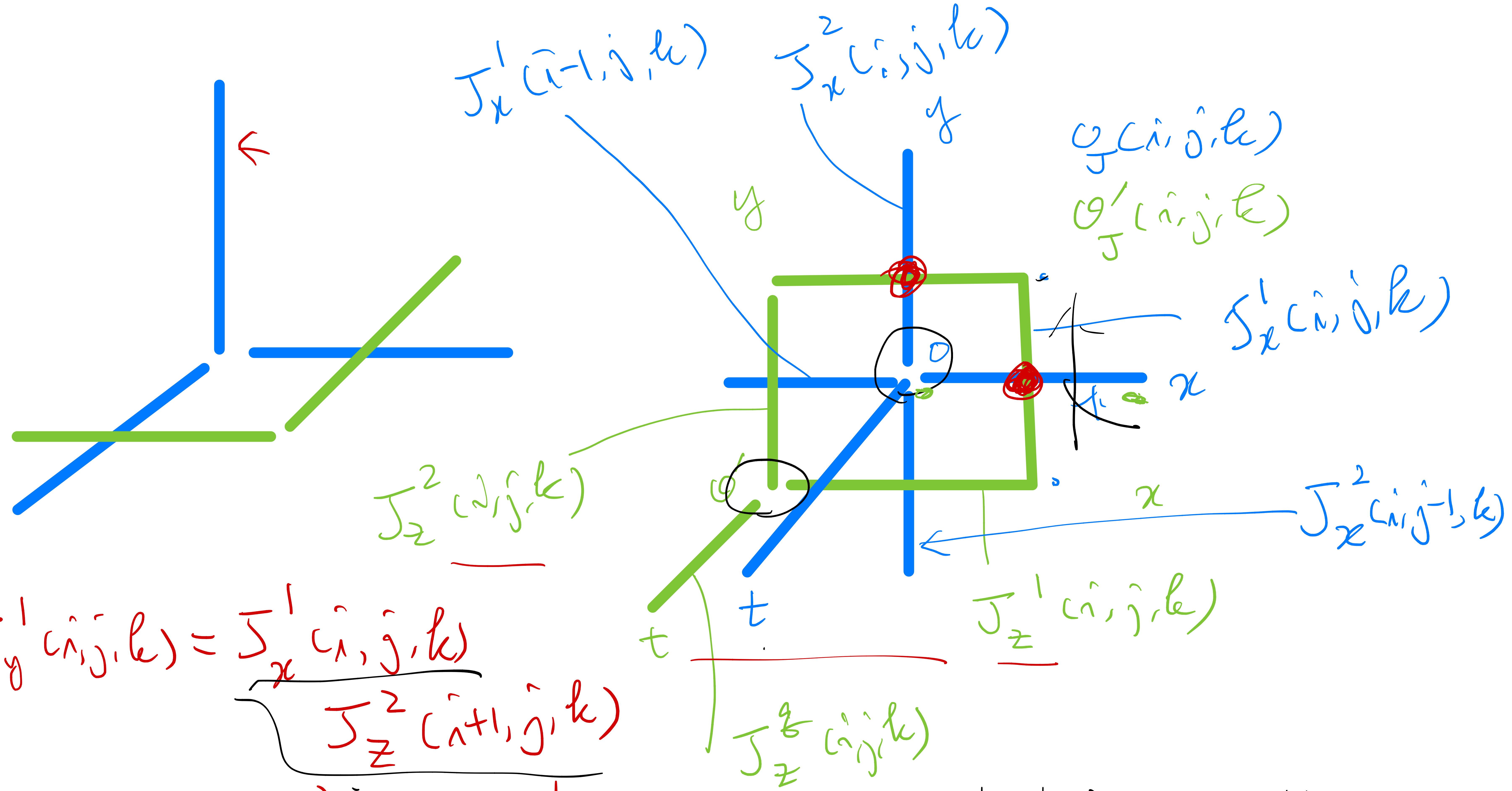
Toric Code circuit unit cell



Qubit error interpretation



Measurement error interpretation



$$J_y^1(i,j,k) = J_x^1(i,j,k)\, J_z^2(i+1,j,k)$$

$$J_y^2(i,j,k) = J_x^2(i,j,k)\, J_z^1(i,j+1,k)$$ ← checked, correct

Lattice hamiltonian for simulation
Phase flip or Bit flip

$$
\begin{aligned}
H_{x,z}(i,j,k) = &- J_x{}^1(i,j,k)\sigma_y(i,j,k)\sigma_t(i,j+1,k)\sigma_y(i,j,k+1)\sigma_t(i,j,k) \\
&- J_x{}^2(i,j,k)\sigma_t(i,j,k)\sigma_x(i,j,k+1)\sigma_t(i+1,j,k)\sigma_x(i,j,k) \\
&- J_x{}^3(i,j,k)\sigma_x(i,j,k)\sigma_y(i+1,j,k)\sigma_x(i,j+1,k)\sigma_y(i,j,k) \\
&- J_z{}^1(i,j,k)\tau_y(i,j,k)\tau_t(i,j+1,k)\tau_y(i,j,k+1)\tau_t(i,j,k) \\
&- J_z{}^2(i,j,k)\tau_t(i,j,k)\tau_x(i,j,k+1)\tau_t(i+1,j,k)\tau_x(i,j,k) \\
&- J_z{}^3(i,j,k)\tau_x(i,j,k)\tau_y(i+1,j,k)\tau_x(i,j+1,k)\tau_y(i,j,k)
\end{aligned}
$$

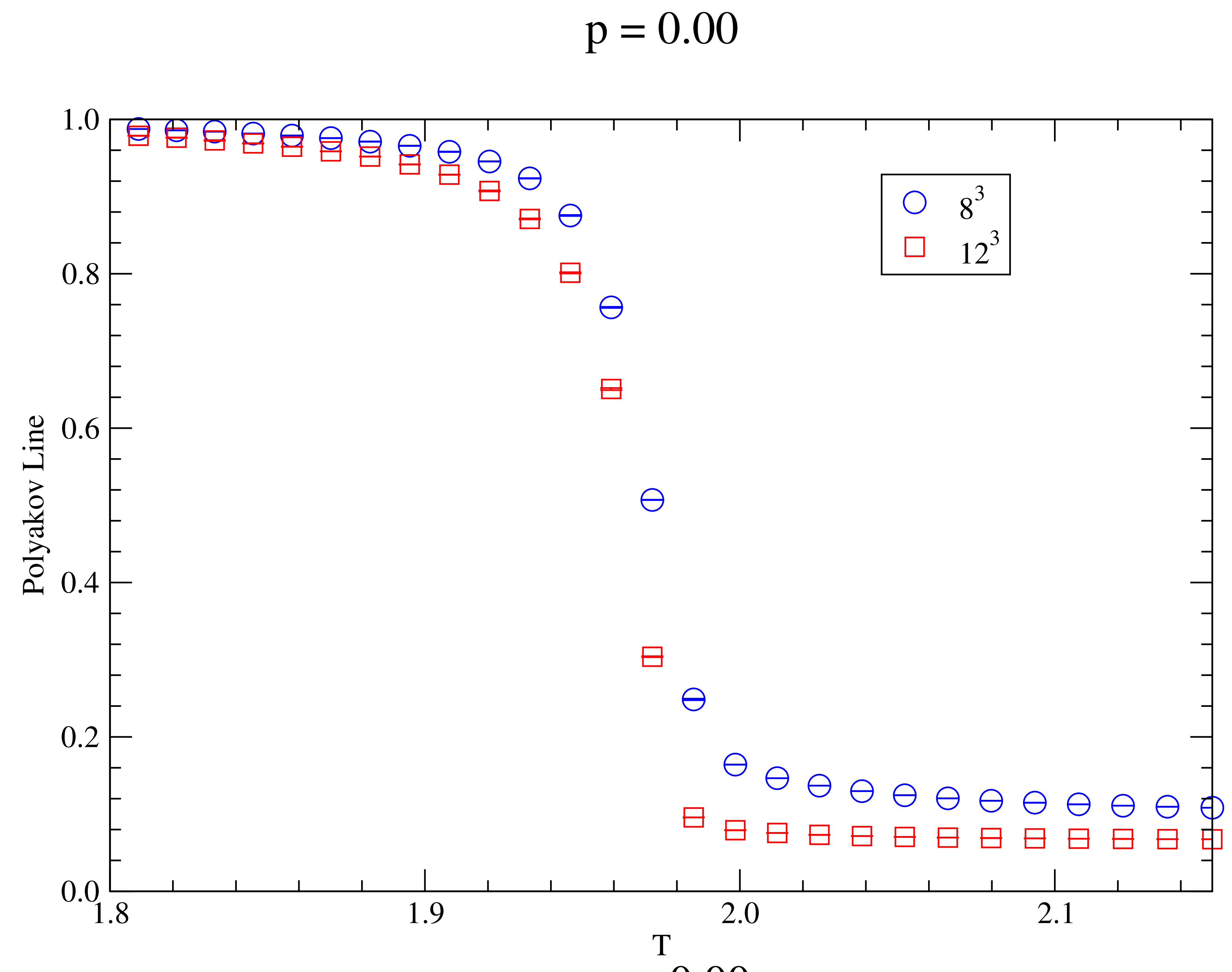
Lattice hamiltonian for simulation
Phase flip and Bit flip together

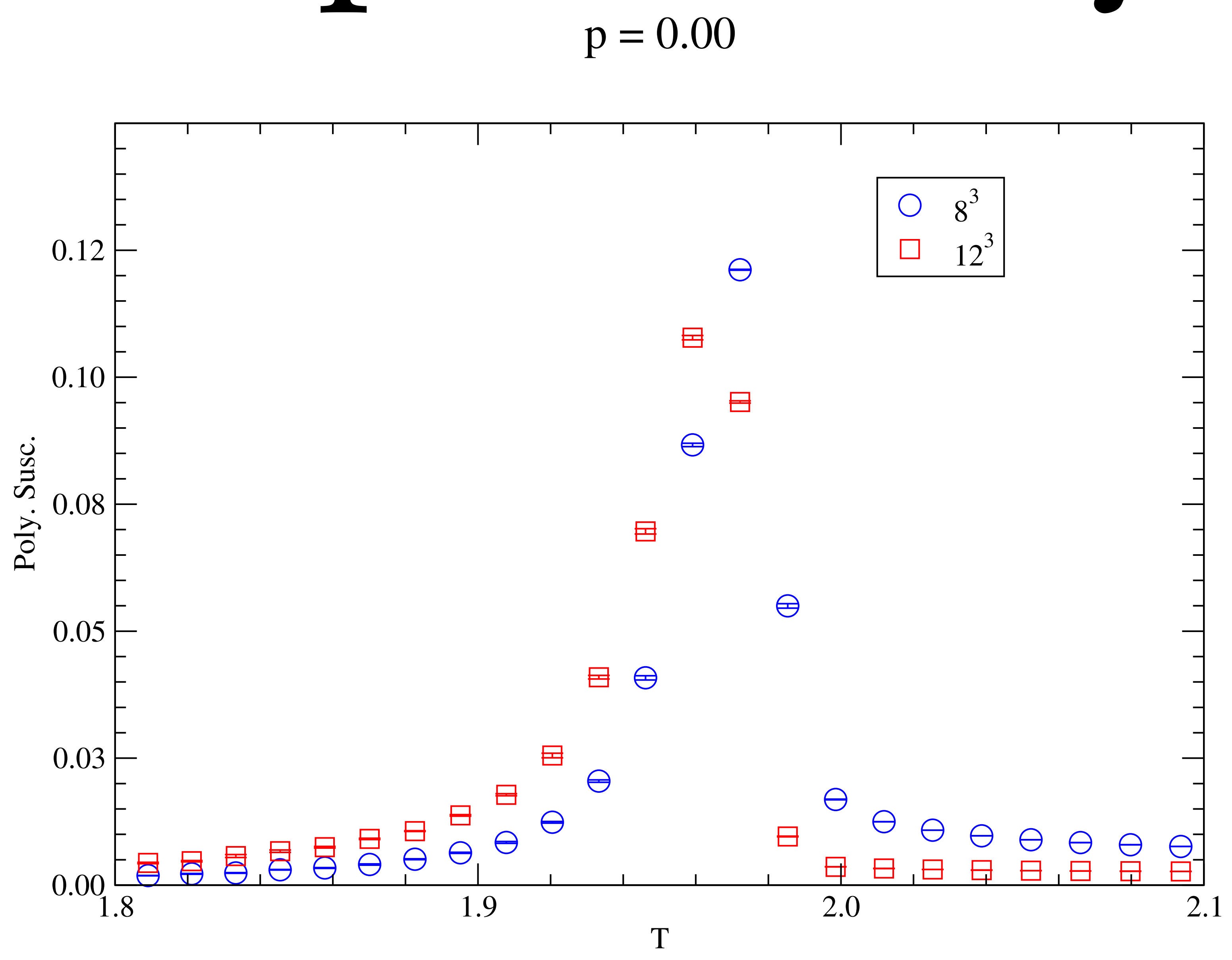
$$
\begin{aligned}
H_y(i,j,k) = -\, J_y{}^1(i,j,k)\sigma_y(i,j,k)\sigma_t(i,j+1,k)\sigma_y(i,j,k+1)\sigma_t(i,j,k) \\
\times\, \tau_t(i+1,j,k)\tau_x(i+1,j,k+1)\tau_t(i+2,j,k)\tau_x(i+1,j,k) \\
-J_y{}^2(i,j,k)\sigma_t(i,j,k)\sigma_x(i,j,k+1)\sigma_t(i+1,j,k)\sigma_x(i,j,k) \\
\times\, \tau_y(i,j+1,k)\tau_t(i,j+2,k)\tau_y(i,j+1,k+1)\tau_t(i,j+1,k)
\end{aligned}
$$

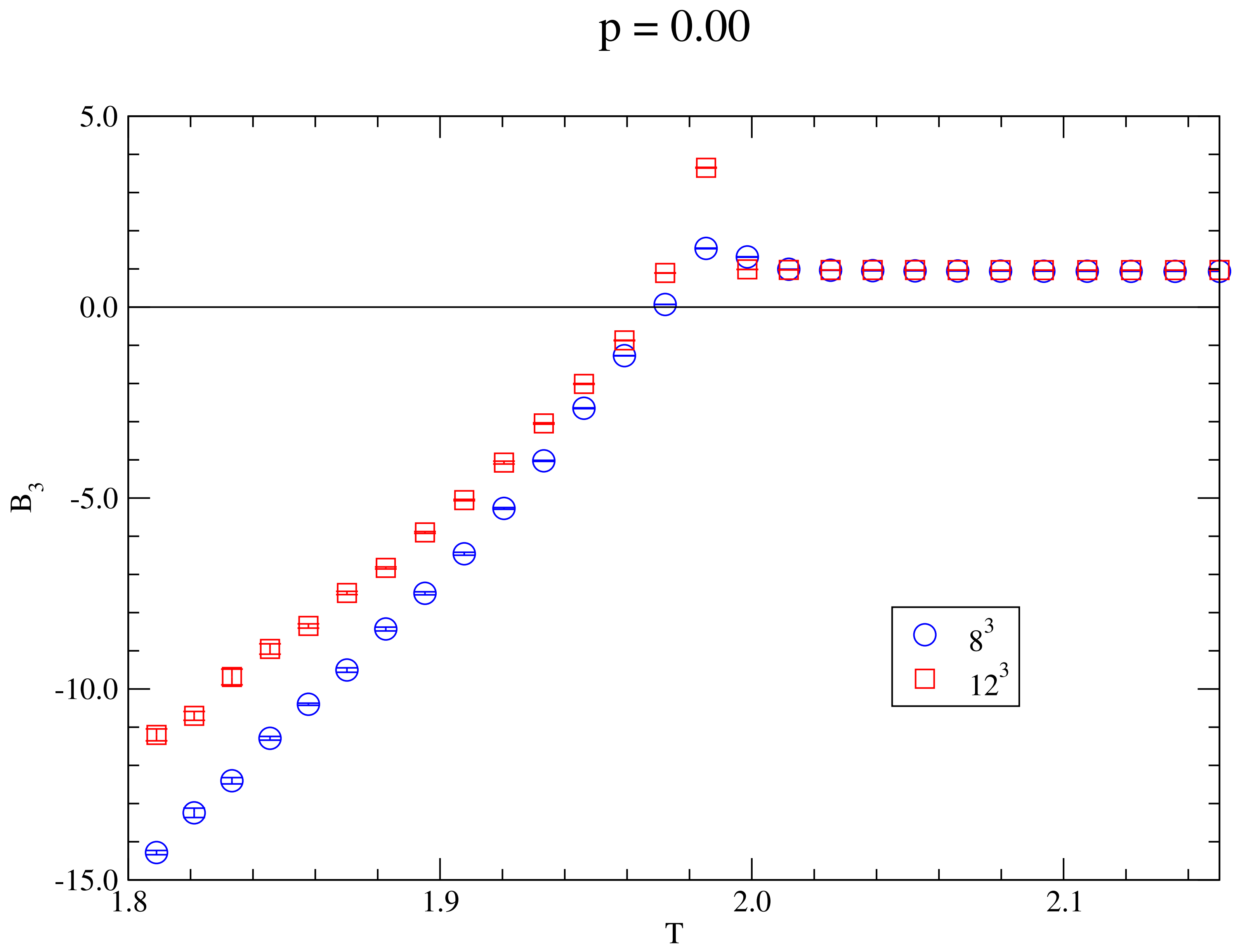
Monte Carlo Simulation

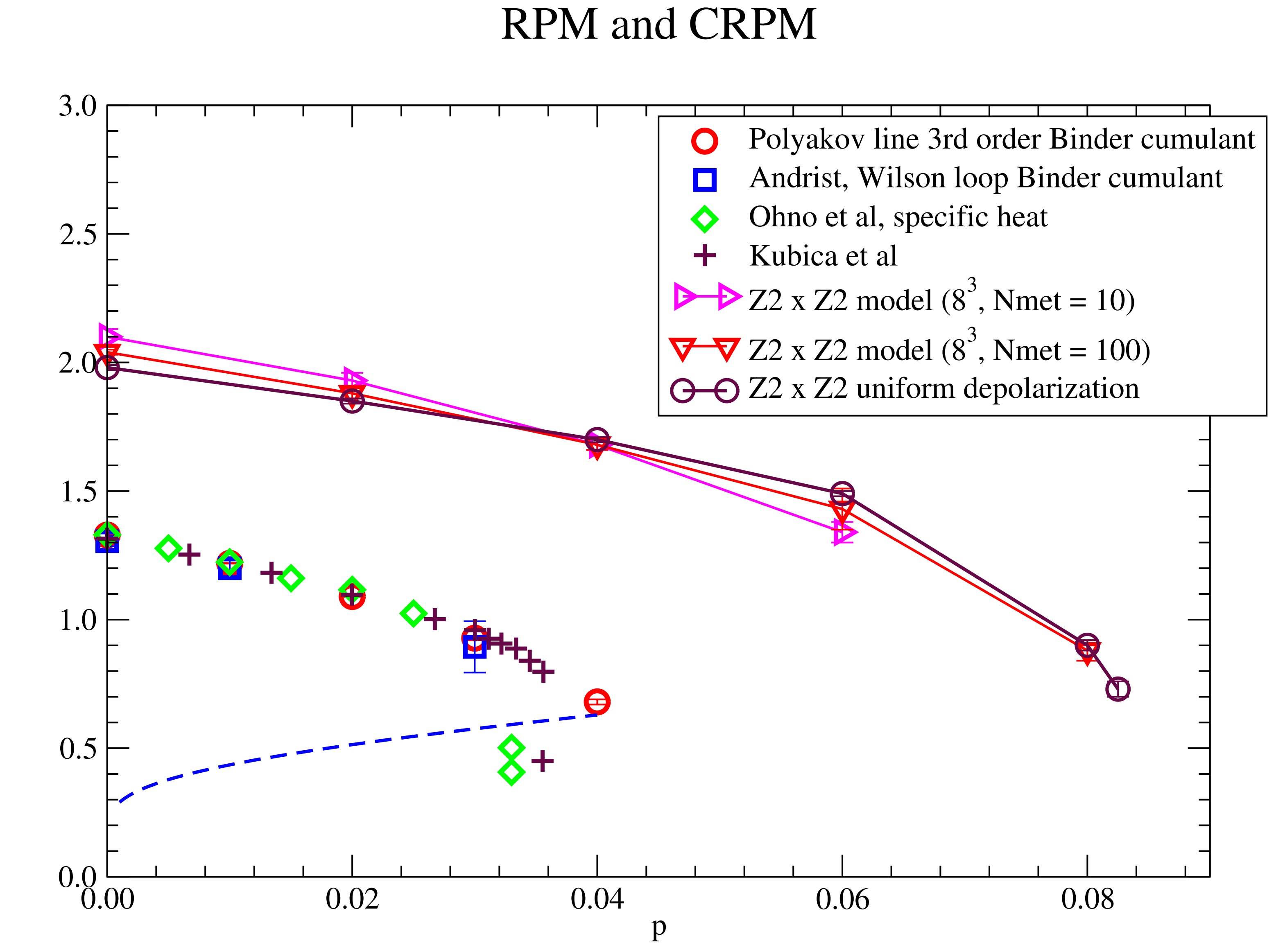
Finding the threshold probability for fault–tolerant quantum computing with a depolarizing noise in which surface code/toric code is used to express logical informations is equivalent to finding the maximum quenching probability above which there is no ferromagnetic phase in quenched $Z(2) \times Z(2)$ gauge theory

Monte Carlo Simulation (preliminary)









cf. R. Acharya et al (Google Quantum AI), "Suppressing quantum errors by scaling a surface code logical qubit", Nature 614 (2023) 676

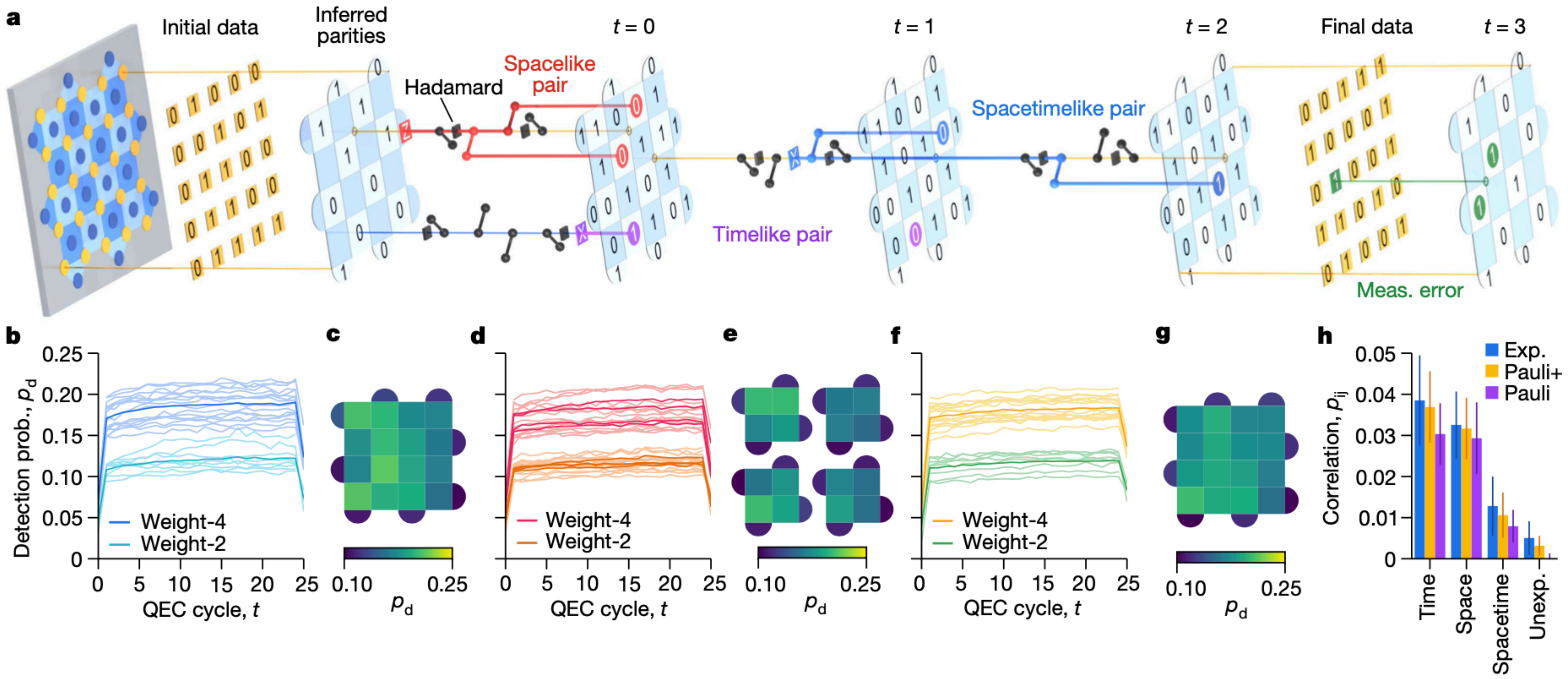


Fig. 2 | Error detection in the surface code. a, Illustration of a surface code experiment, in perspective view with time progressing to the right. We begin with an initial data qubit state that has known parities in one stabilizer basis (here, Z). We show example errors that manifest in detection pairs: a Z error (red) on a data qubit (spacelike pair), a measurement error (purple) on a measure qubit (timelike pair), an X error (blue) during the CZ gates (spacetimelike pair) and a measurement error (green) on a data qubit (detected in the final inferred Z parities). **b**, Detection probability for each stabilizer over a 25-cycle distance-5 experiment (50,000 repetitions). Darker lines: average over all stabilizers with the same weight. There are fewer detections at timestep $t = 0$ because there is no preceding syndrome extraction, and at $t = 25$ because the final parities are calculated from data qubit measurements directly. QEC, quantum error correction. **c**, Detection probability heatmap, averaging over $t = 1$ to 24. **d**,**e**, Similar to **b**,**c** for four separate distance-3 experiments covering the four quadrants of the distance-5 code. **f**,**g**, Similar to **b**,**c** using a simulation with Pauli errors plus leakage, crosstalk and stray interactions (Pauli+). **h**, Bar chart summarizing the detection correlation matrix p_{ij}, comparing the distance-5 experiment from **b** to the simulation in **f** (Pauli+) and a simpler simulation with only Pauli errors. We aggregate four groups of correlations: timelike pairs; spacelike pairs; spacetimelike pairs expected for Pauli noise; and spacetimelike pairs unexpected for Pauli noise (Unexp.), including correlations over two timesteps. Each bar shows a mean and standard deviation of correlations from a 25-cycle, 50,000-repetition dataset.

cf. R. Acharya et al (Google Quantum AI), "Suppressing quantum errors by scaling a surfac code logical qubit", Nature 614 (2023) 676

Article

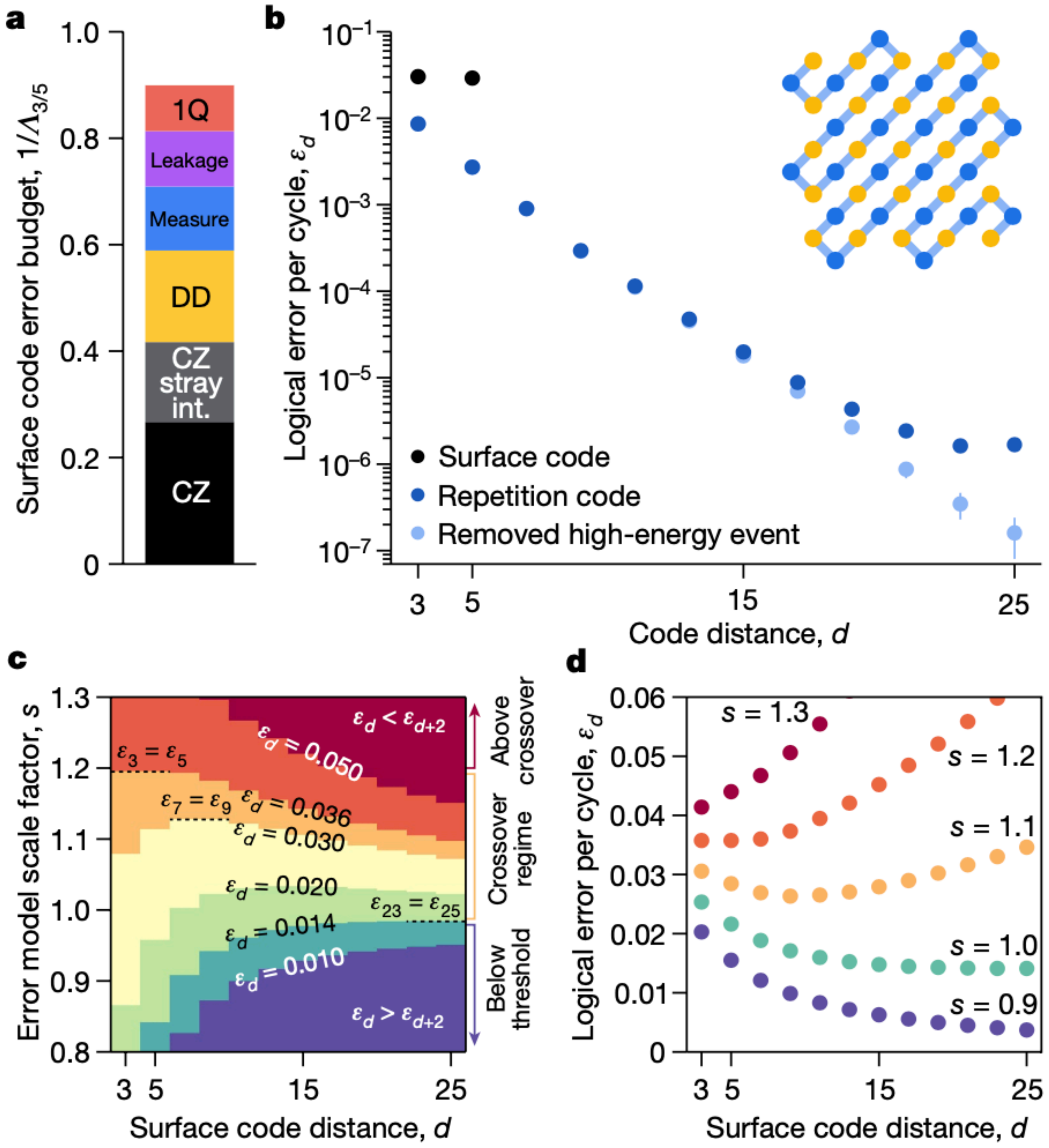


Fig. 4 | Towards algorithmically relevant error rates. a, Estimated error budget for the surface code, based on component errors (see Fig. 1c) and Pauli+ simulations. $\Lambda_{3/5} = \varepsilon_3/\varepsilon_5$. CZ, contributions from CZ error (excluding leakage and stray interactions). CZ stray int., CZ error from unwanted interactions. DD, dynamical decoupling (data qubit idle error during measurement and reset). Measure, measurement and reset error. Leakage, leakage during CZs and due to heating. 1Q, single-qubit gate error. **b**, Logical error for repetition codes.

4. Discussions

- “Threshold probability”, the largest error rate below which fault-tolerant quantum computing is possible can be investigated through QEC – Statistical Model mapping
- We show that for the depolarizing noise and the measurement error from a realistic quantum circuits of QEC for the surface code, the mapped statistical model is quenched Z(2) x Z(2) gauge theory in the 3-dimension
- To find the threshold probability of this model, Monte Carlo simulation of the chosen statistical model using parallel tempering and metropolis updates is currently being performed

Stay tuned!

Supplementary

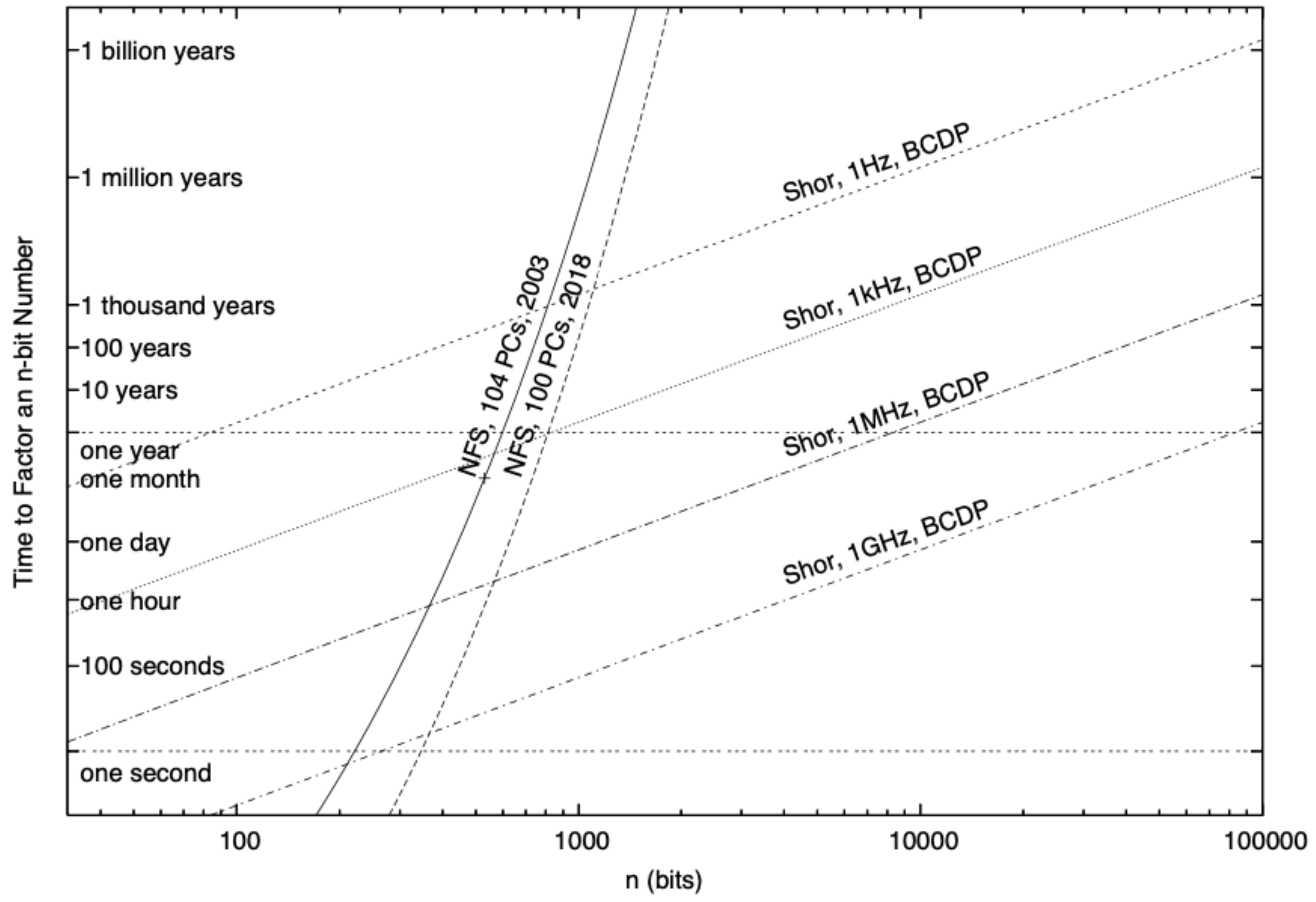
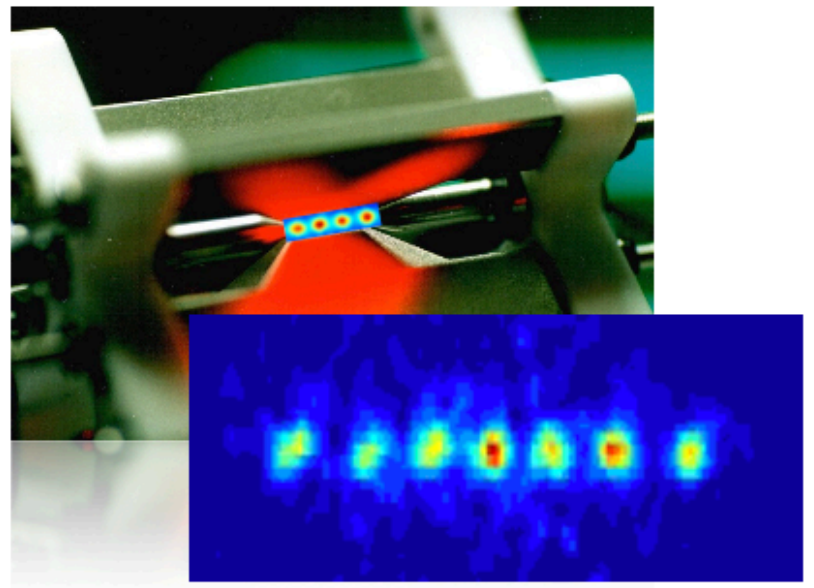


Figure 1. Scaling of number field sieve (NFS) on classical computers and Shor's algorithm for factoring on a quantum computer, using BCDP modular exponentiation with various clock rates. Both horizontal and vertical axes are log scale. The horizontal axis is the size of the number being factored.

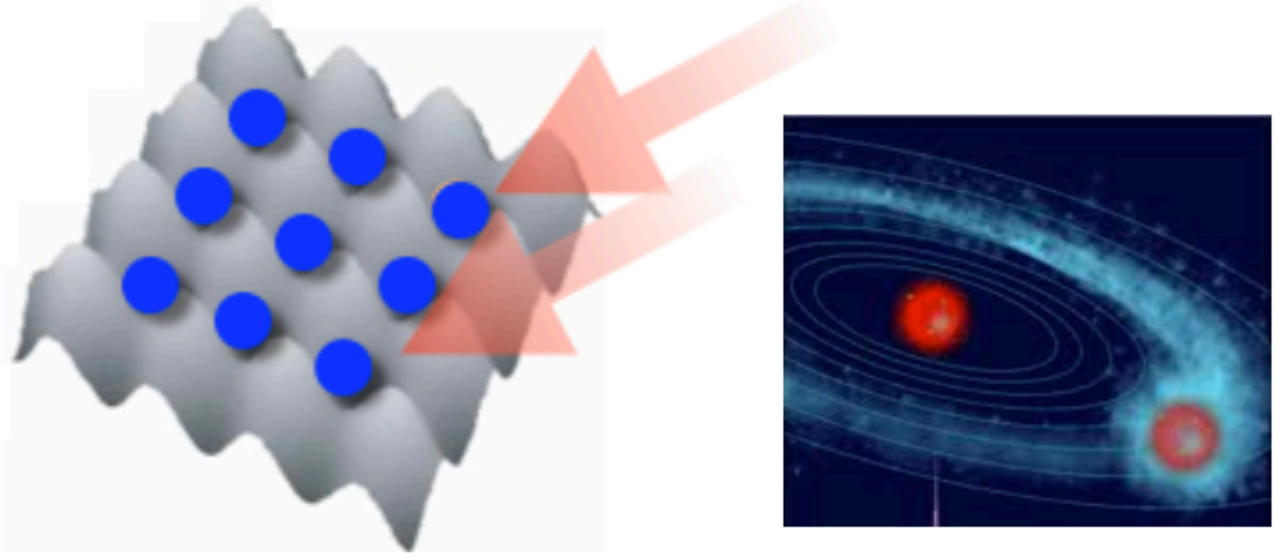
R. Van Meter et al., arXiv:quant-ph/0507023

Physical systems

Trapped ions

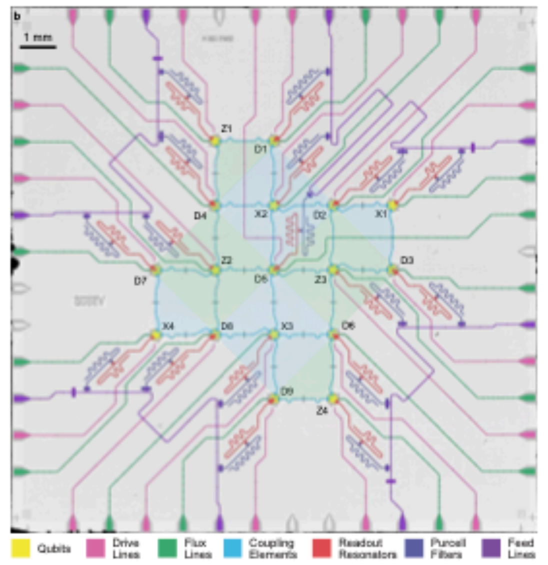


Neutral & Rydberg atoms

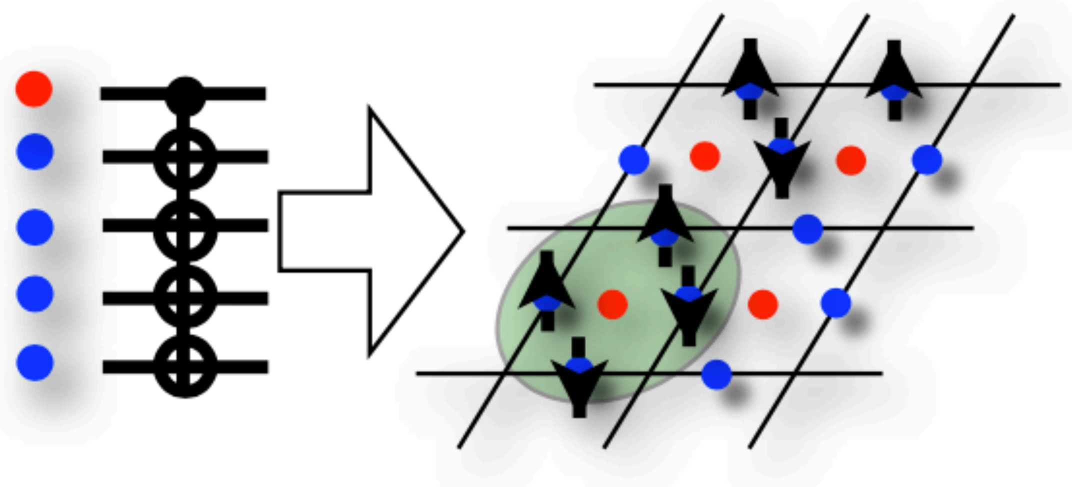


Rydberg ions

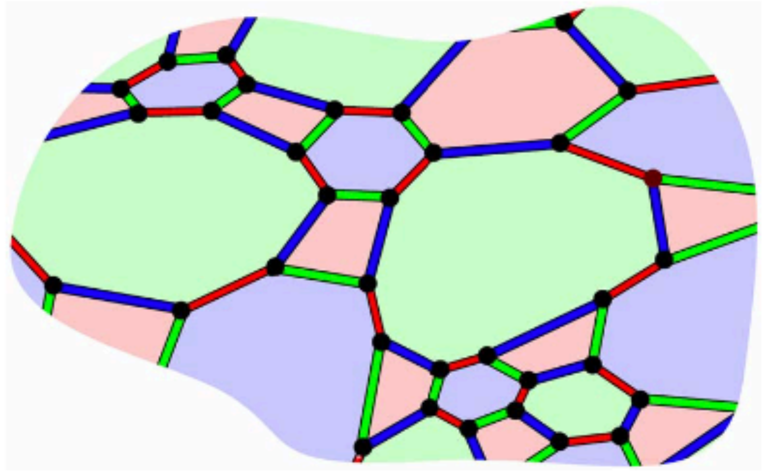
Supercond. qubits



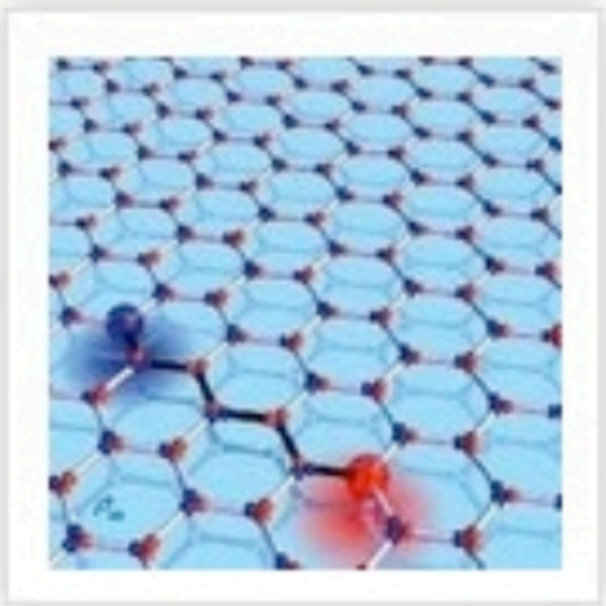
Main Research Lines



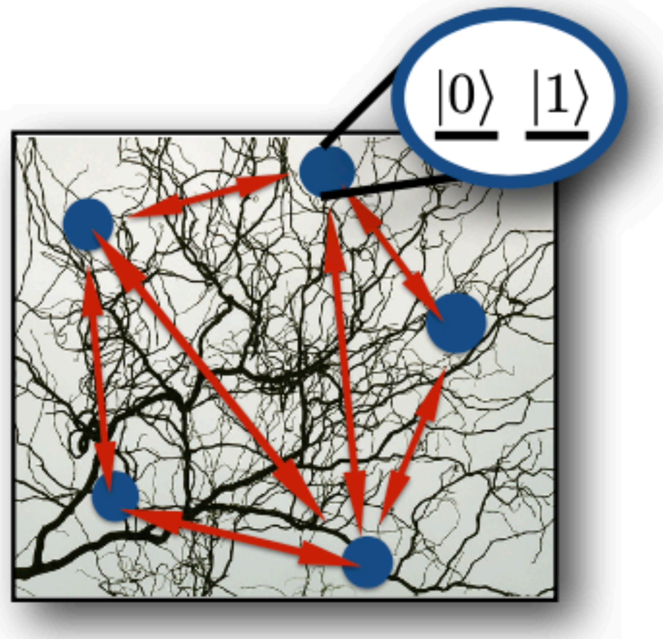
Quantum Computing
AMO System Engineering
Microscopic Modelling
Quantum Characterisation



Quantum Error Correction
Fault-tolerance, Decoders
Topological Codes
Realistic Protocols & Feasibility



Quantum Simulation
Reservoir Engineering
Topological Q. Matter



Quantum Neural Networks
Open-System
Quantum Networks
Q. Machine Learning

www.rwth-aachen.de/mueller-group

www.markus-mueller.website