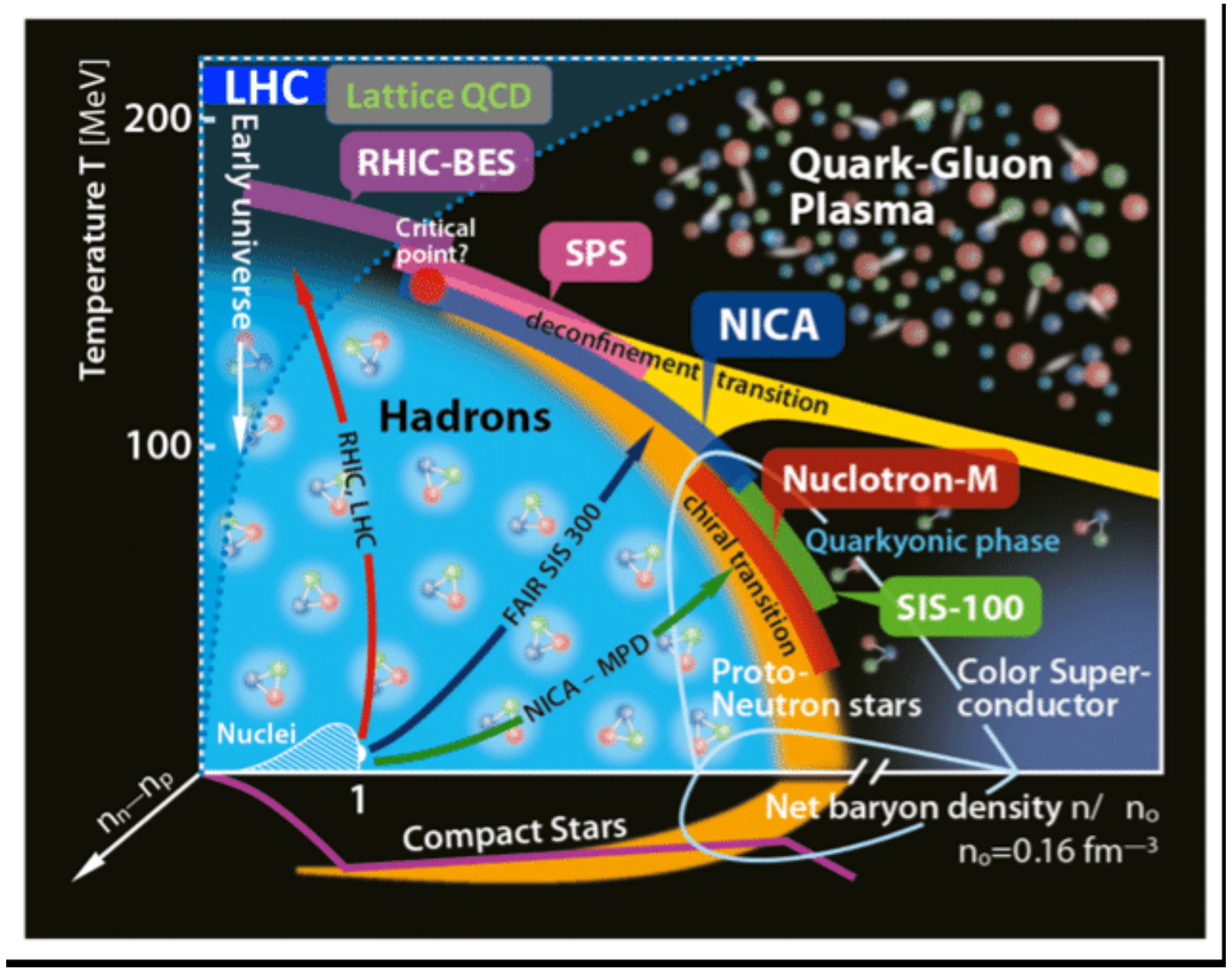
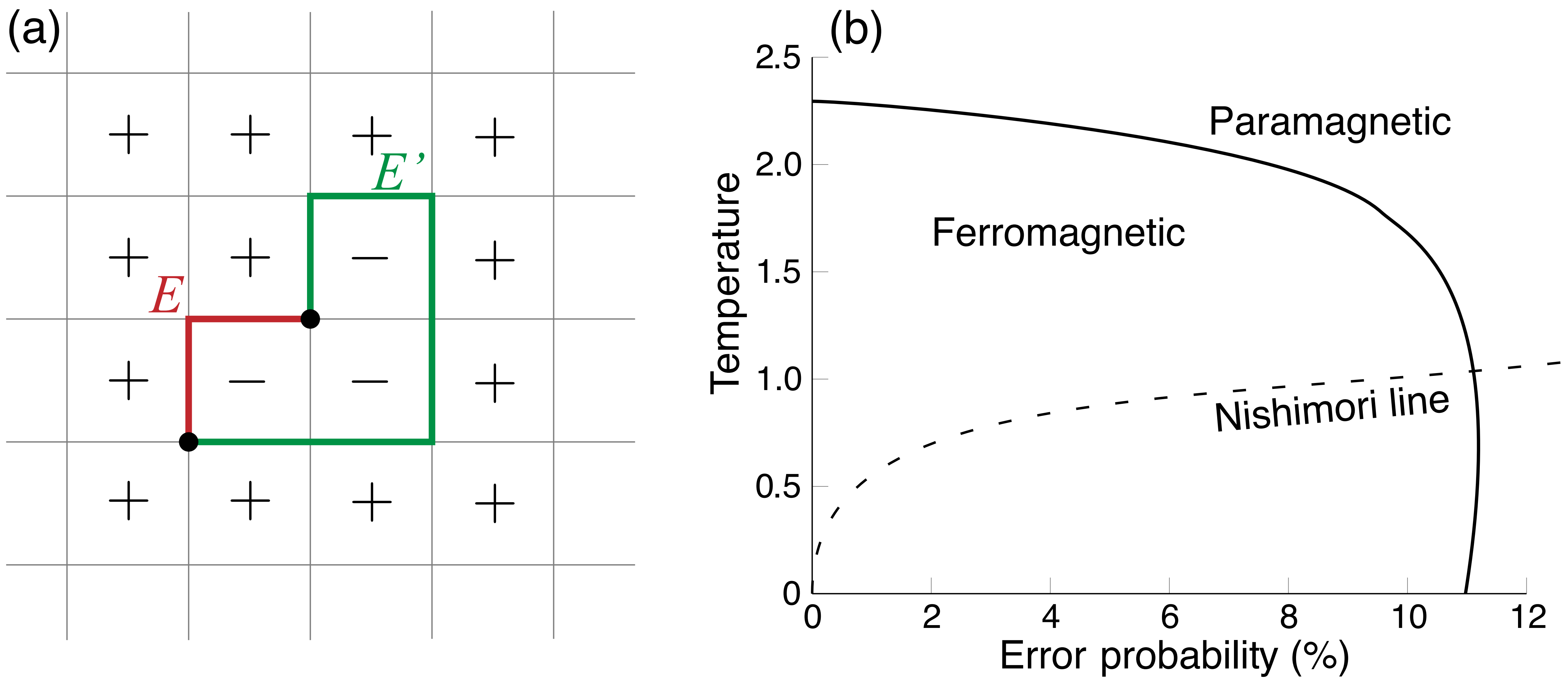
Viability of Error Correction in Quantum Computer and Z(2) Gauge Theory

Seyong Kim
Sejong University

In collaboration with M. Mueller (IQI, RWTH), M. Rispler (IQI, RWTH) and D. Vodola (BASF)
Quantum 6 (2022) 618, and arXiv:2412:14004 (accepted in Nature Physics Communication Quantum)



arXiv.2201.00202



arXiv.2412.14004

$$< \mathcal{O} > = \frac{\int dU_\mu e^{-S_E[U]} \det(\mathcal{M}[U, m_q])\mathcal{O}[U]}{\int dU_\mu e^{-S_E[U]} \det(\mathcal{M}[U, m_q])} = \int dU_\mu P[U_\mu, m_q]\mathcal{O}[U]$$

$$\mathcal{M}[U, m_q] = \gamma_\mu^E D_\mu + m_q \rightarrow \quad \gamma_5 \mathcal{M}[U, m_q]\gamma_5 = -\gamma_\mu^E D_\mu + m_q = \gamma_\mu^E D_\mu^\dagger + m_q$$

$$\det\left(\mathcal{M}[U, m_q]\right) = \sqrt{\det\left(\mathcal{M}[U, m_q]\right)\det\left(\mathcal{M}[U, m_q]\right)} = \sqrt{\det\left(\mathcal{M}[U, m_q]\right)\det\left(\gamma_5 \mathcal{M}[U, m_q]\gamma_5\right)}$$

$$= \sqrt{\det\left(\mathcal{M}\mathcal{M}^\dagger\right)}$$

With baryon chemical potential, can’t be done → complex action problem

$$< \mathcal{O} > = \frac{\int dU_\mu e^{-S_E[U]} \det(\mathcal{M}[U, m_q])\mathcal{O}[U]}{\int dU_\mu e^{-S_E[U]} \det(\mathcal{M}[U, m_q])} = \int dU_\mu P[U_\mu, m_q]\mathcal{O}[U]$$

$$\mathcal{M}[U, m_q, \mu_q] = \gamma_\mu^E D_\mu + m_q + \mu_q \gamma_4^E \rightarrow$$

$$\gamma_5 \mathcal{M}[U, m_q, \mu_q]\gamma_5 = -\gamma_\mu^E D_\mu + m_q - \mu_q \gamma_4^E = \gamma_\mu^E D_\mu^\dagger + m_q - \mu_q \gamma_4^E = \mathcal{M}[U, m_q, -\mu_q^*]^\dagger$$

$$\mathcal{M}[U, m_q, \mu_q] \neq \mathcal{M}[U, m_q, \mu_q]^\dagger$$

$$\frac{e^{-S_E[U]} \det(\mathcal{M}[U, m_q])}{\int dU_\mu e^{-S_E[U]} \det(\mathcal{M}[U, m_q])}$$

Can't be interpreted as a probability weight

PRL **94**, 170201 (2005) PHYSICAL REVIEW LETTERS week ending 6 MAY 2005

Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations

Matthias Troyer[1] and Uwe-Jens Wiese[2]

[1]*Theoretische Physik, ETH Zürich, CH-8093 Zürich, Switzerland*

[2]*Institut für theoretische Physik, Universität Bern, Sidlerstrasse 5, CH-3012 Bern, Switzerland*

(Received 11 August 2004; published 4 May 2005)

Quantum Monte Carlo simulations, while being efficient for bosons, suffer from the ‘‘negative sign problem’’ when applied to fermions—causing an exponential increase of the computing time with the number of particles. A polynomial time solution to the sign problem is highly desired since it would provide an unbiased and numerically exact method to simulate correlated quantum systems. Here we show that such a solution is almost certainly unattainable by proving that the sign problem is nondeterministic polynomial (NP) hard, implying that a generic solution of the sign problem would also solve all problems in the complexity class NP in polynomial time.

DOI: 10.1103/PhysRevLett.94.170201 PACS numbers: 02.70.Ss, 05.10.Ln

cond-mat/0408730

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

Last December


YouTube KR
Search
Create
이번 주 초에 잠시 주춤한 비트코인
That is 6.4% below the all-time high of $103,650, established five days ago.
12월 5일 10만 3650달러로 사상 최고치를 찍었던 비트코인은 12월 10일 9만 7천 달러선까지 하락했다.
YouTube 슈카월드
작은칼날부리
아 씨 뭐지? ㅋㅋ
솜 이불
양자컴?
softnatae
코 인
Avatara
양자컴퓨터?
James Din
다 내쫓아야지
흔냥이
팔아요?
투자대박가즈아
구글 양자
KOREA FOX
제일 궁금한 주제
SR
존문가
T R I
큰거온다
CMV Factory
UFO 얘기는 안하나??
Tesla Nikola
10만 3천 달러를 찍었다가
0:32 / 31:45 • 구글의 특급 발표
구글의 양자 칩 윌로우, 암호화폐는 무력화될까?
슈카월드
3.46M subscribers
Subscribed
11K
Share
Thanks
All
From 슈카월드
Cryptocurrency
Politics
$4K/Day Passive Income Arbitrage AI Bot
MysteryCake
22K views
Sponsored
한국경제 유동성 함정에 빠졌습니다 (미즈호은행 변정규 전무)
언더스탠딩 : 세상의 모든 지식
61K views • 11 hours ago
New
50:42
Physicists Say They Know How Cold Fusion Works
Sabine Hossenfelder
166K views • 8 hours ago
New
7:38
Starship Launch IMMINENT: What's wrong with SpaceX's...
What about it!?
279K views • 10 hours ago
New
21:20
수출 역대 최대 기록, 1000조원 달성
슈카월드
656K views • 3 days ago
New
35:05
LA 비상사태 선포, 꺼지지 않는 초대형 산불
슈카월드


Willow's
Superconducting
Qubits
Willow's
Qubit Grid
105 Qubits

nature

https://doi.org/10.1038/s41586-024-08449-y

Accelerated Article Preview

Quantum error correction below the surface code threshold

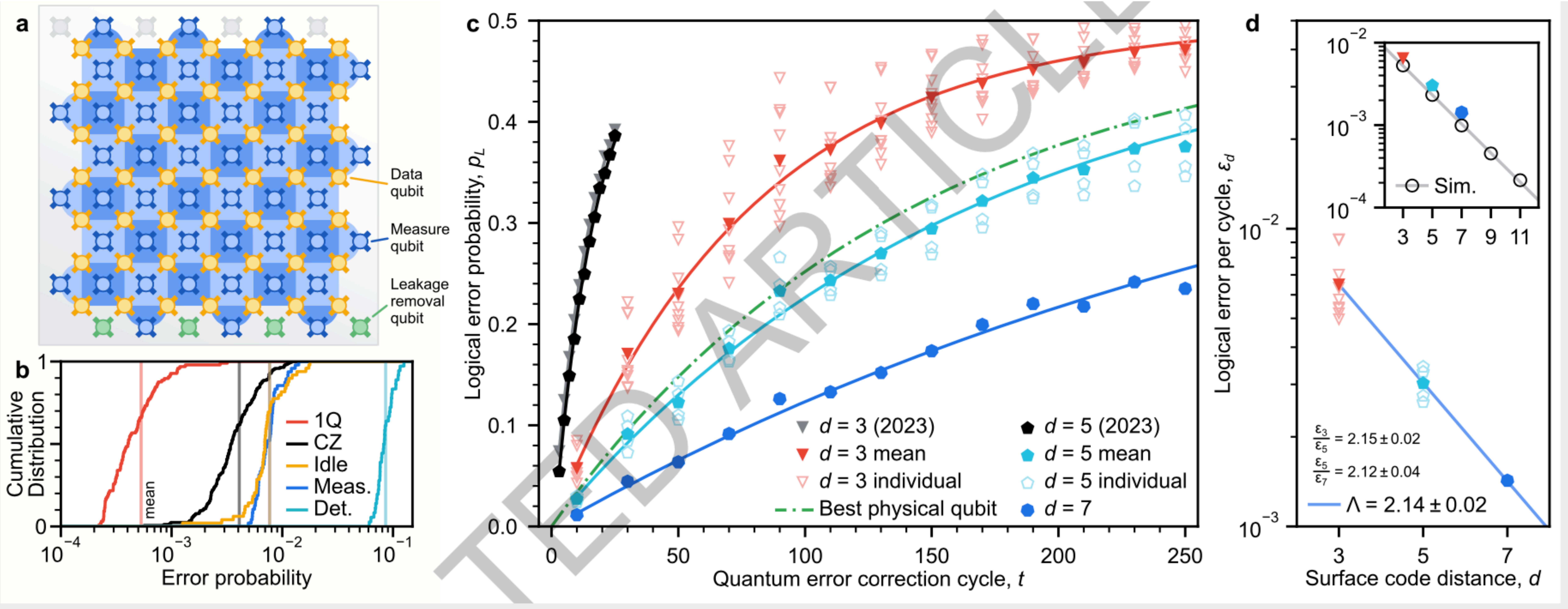
Received: 24 August 2024

Accepted: 25 November 2024

Accelerated Article Preview

Cite this article as: Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature* https://doi.org/10.1038/s41586-024-08449-y (2024)

Google Quantum AI and Collaborators


a
Data qubit
Measure qubit
Leakage removal qubit
b
Cumulative Distribution
mean
1Q
CZ
Idle
Meas.
Det.
Error probability
c
Logical error probability, p_L
Quantum error correction cycle, t
d = 3 (2023)
d = 3 mean
d = 3 individual
Best physical qubit
d = 5 (2023)
d = 5 mean
d = 5 individual
d = 7
d
Logical error per cycle, ε_d
Sim.
$\frac{\varepsilon_3}{\varepsilon_5} = 2.15 \pm 0.02$
$\frac{\varepsilon_5}{\varepsilon_7} = 2.12 \pm 0.04$
$\Lambda = 2.14 \pm 0.02$
Surface code distance, d

Example of QC: Shor's algorithm

Algorithms for Quantum Computation: Discrete Logarithms and Factoring

Peter W. Shor
AT&T Bell Labs
Room 2D-149
600 Mountain Ave.
Murray Hill, NJ 07974, USA

Abstract

A computer is generally considered to be a universal computational device; i.e., it is believed able to simulate any physical computational device with a cost in computation time of at most a polynomial factor. It is not clear whether this is still true when quantum mechanics is taken into consideration. Several researchers, starting with David Deutsch, have developed models for quantum mechanical computers and have investigated their computational properties. This paper gives Las Vegas algorithms for finding discrete logarithms and factoring integers on

[1, 2]. Although he did not ask whether quantum mechanics conferred extra power to computation, he did show that a Turing machine could be simulated by the reversible unitary evolution of a quantum process, which is a necessary prerequisite for quantum computation. Deutsch [9, 10] was the first to give an explicit model of quantum computation. He defined both quantum Turing machines and quantum circuits and investigated some of their properties.

The next part of this paper discusses how quantum computation relates to classical complexity classes. We will thus first give a brief intuitive discussion of complexity classes for those readers who do not have this background.

Rivest–Shamir–Adleman (RSA) algorithm

RSA consists of 4 steps:

(1)key generation (public, private)

(2)key distribution

(3)encoding

(4)decoding

Example of RSA: key generation

1. Choose two prime numbers (p, q) (e.g. $p = 61, q = 53$)
2. Compute $n = pq$ (e.g. $n = pq = 61 \times 53 = 3233$)
3. Compute Carmichael's totient function $\lambda(n) = lcm(p-1, q-1)$ (e.g. $\lambda(n) = lcm(p-1, q-1) = lcm(60,52) = 780$)
4. Choose any number $0 < e < \lambda(n)$ that is coprime to $\lambda(n)$ (choose a prime number that is not a divisor of $\lambda(n)$, e.g. $e = 17$)
5. Compute d, the modular multiplicative inverse of $e(\mod \lambda(n))$ (for $e = 17$, $\lambda(n) = 780$, $d = 413$ since $1 = 17 \times 413 \mod (780)$)
6. Public key is (n, e) and private key is (n, d) (for our example, $(n, e) = (3233,17)$, $(n, d) = (3233,413)$)

Example of RSA: encoding and decoding

1. Encryption for the message, m is

   $c(m) = m^e \mod (n)$ (e.g, $m^{17} \mod (3233)$)

2. Decryption for the message, m is

   $m(c) = c^d \mod (n)$ (e.g, $c^{413} \mod (3233)$)

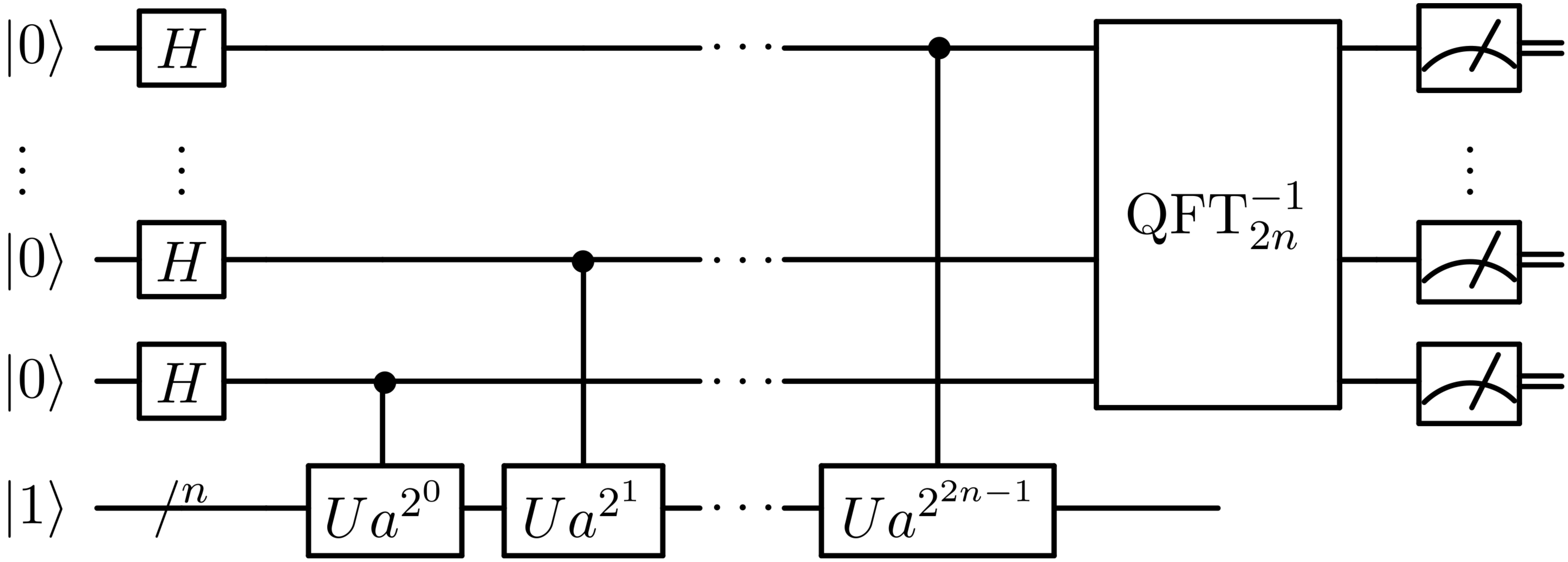
Shor's algorithm: Find the prime factors

1. Choose a random number a, $1 < a < N$
2. Compute $K = gcd(a, N)$
3. If $K \neq 1$, K is a nontrivial factor. The other factor is $\frac{N}{K}$
4. If not, use quantum routine to find the order r of a: find r such that $a^r = 1 \mod N$
5. If r is odd, go back to the step 1
6. Compute $g = gcd(N, a^{\frac{r}{2}} + 1)$. If g is nontrivial, the other factor is $\frac{N}{g}$. Otherwise go to the step 1

Shor's algorithm: Quantum subroutine arXiv:quant-ph/9508027

1. Use quantum phase estimation with unitary operator U (this represents multiplication by a) on $|0\rangle^{\otimes 2n} \otimes |1\cdots1\rangle$ (2nd register has n qubits)
2. Use classical computer to find r from the measurement outcomes

Shor's algorithm: Quantum subroutine



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

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

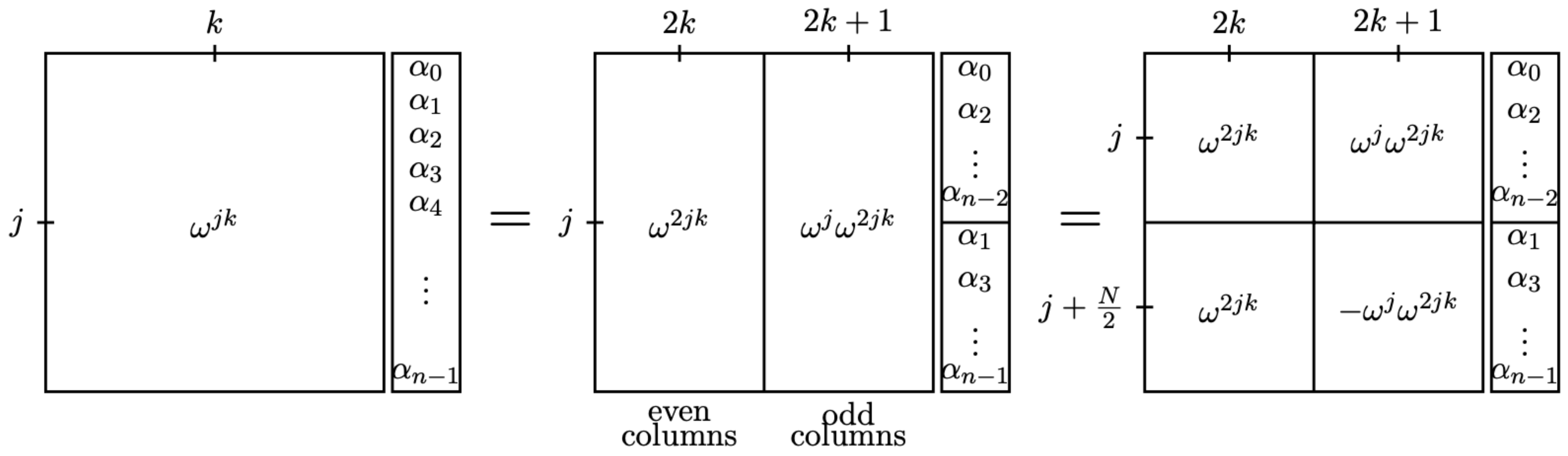
For $k = 1, \cdots, N,$

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

Cooley–Tukey Algorithm

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_8^1 & \omega_8^2 & \omega_8^3 & \omega_8^4 & \omega_8^5 & \omega_8^6 & \omega_8^7 \\ 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 & 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 \\ 1 & \omega_8^3 & \omega_8^6 & \omega_8^1 & \omega_8^4 & \omega_8^7 & \omega_8^2 & \omega_8^5 \\ 1 & \omega_8^4 & 1 & \omega_8^4 & 1 & \omega_8^4 & 1 & \omega_8^4 \\ 1 & \omega_8^5 & \omega_8^2 & \omega_8^7 & \omega_8^4 & \omega_8^1 & \omega_8^6 & \omega_8^3 \\ 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 & 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 \\ 1 & \omega_8^7 & \omega_8^6 & \omega_8^5 & \omega_8^4 & \omega_8^3 & \omega_8^2 & \omega_8^1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix}$$

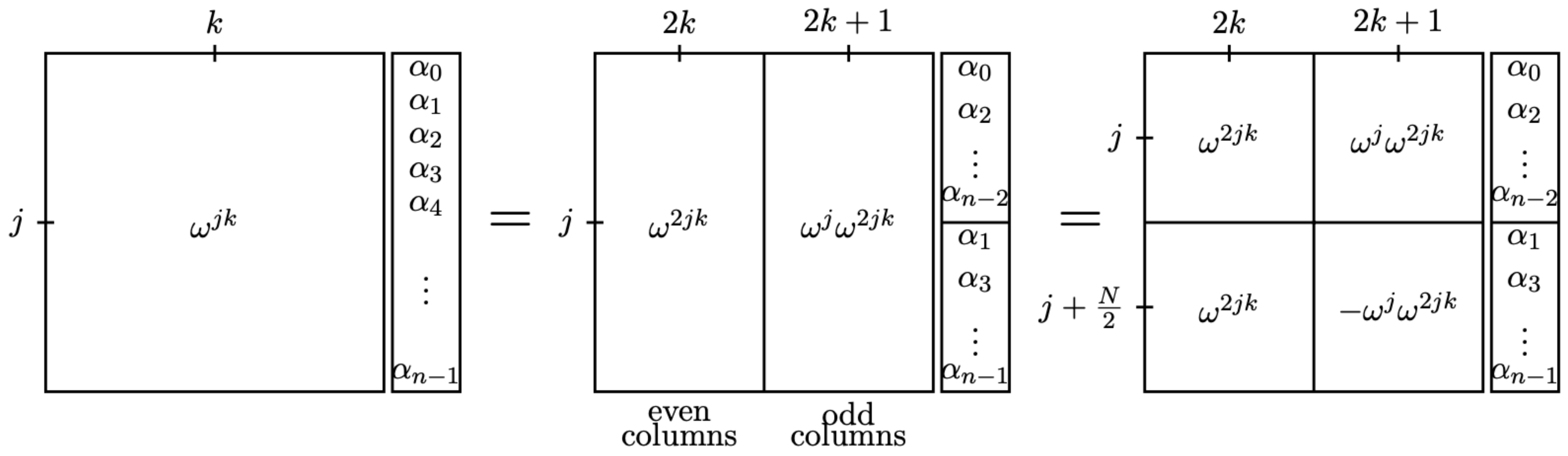
$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8 & \omega_8^3 & \omega_8^5 & \omega_8^7 \\ 1 & \omega_8^4 & 1 & \omega_8^4 & \omega_8^2 & \omega_8^6 & \omega_8^2 & \omega_8^6 \\ 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^3 & \omega_8^1 & \omega_8^7 & \omega_8^5 \\ 1 & 1 & 1 & 1 & \omega_8^4 & \omega_8^4 & \omega_8^4 & \omega_8^4 \\ 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^5 & \omega_8^7 & \omega_8^1 & \omega_8^3 \\ 1 & \omega_8^4 & 1 & \omega_8^4 & \omega_8^6 & \omega_8^2 & \omega_8^6 & \omega_8^2 \\ 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^7 & \omega_8^5 & \omega_8^3 & \omega_8^1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_2 \\ x_4 \\ x_6 \\ x_1 \\ x_3 \\ x_5 \\ x_7 \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix}$$



Cooley–Tukey Algorithm

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8 & \omega_8^3 & \omega_8^5 & \omega_8^7 \\ 1 & \omega_8^4 & 1 & \omega_8^4 & \omega_8^2 & \omega_8^6 & \omega_8^2 & \omega_8^6 \\ 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^3 & \omega_8^1 & \omega_8^7 & \omega_8^5 \\ 1 & 1 & 1 & 1 & \omega_8^4 & \omega_8^4 & \omega_8^4 & \omega_8^4 \\ 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^5 & \omega_8^7 & \omega_8^1 & \omega_8^3 \\ 1 & \omega_8^4 & 1 & \omega_8^4 & \omega_8^6 & \omega_8^2 & \omega_8^6 & \omega_8^2 \\ 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^7 & \omega_8^5 & \omega_8^3 & \omega_8^1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_2 \\ x_4 \\ x_6 \\ x_1 \\ x_3 \\ x_5 \\ x_7 \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8 & \omega_8^3 & \omega_8^5 & \omega_8^7 \\ 1 & \omega_8^4 & 1 & \omega_8^4 & \omega_8^2 & \omega_8^6 & \omega_8^2 & \omega_8^6 \\ 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^3 & \omega_8^1 & \omega_8^7 & \omega_8^5 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & \omega_8^2 & \omega_8^4 & \omega_8^6 & -\omega_8 & -\omega_8^3 & -\omega_8^5 & -\omega_8^7 \\ 1 & \omega_8^4 & 1 & \omega_8^4 & -\omega_8^2 & -\omega_8^6 & -\omega_8^2 & -\omega_8^6 \\ 1 & \omega_8^6 & \omega_8^4 & \omega_8^2 & -\omega_8^3 & -\omega_8^1 & -\omega_8^7 & -\omega_8^5 \end{pmatrix} \begin{pmatrix} x_0 \\ x_2 \\ x_4 \\ x_6 \\ x_1 \\ x_3 \\ x_5 \\ x_7 \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix}$$



$$\omega_8^4 = (e^{\frac{2\pi i}{8}})^4 = e^{i\pi} = -1$$

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

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

For Quantum computer,

the number of operations
for Fast Fourier transform is

$$\sim n^2$$

Quantum Fourier transform

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

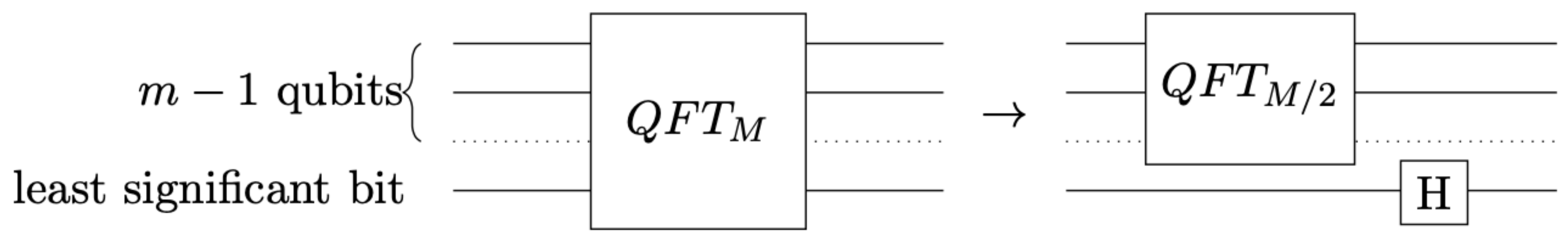


Figure 5.4: $QFT_{M/2}$ and a Hadamard gate correspond to $FFT_M/2$ on the odd and even terms

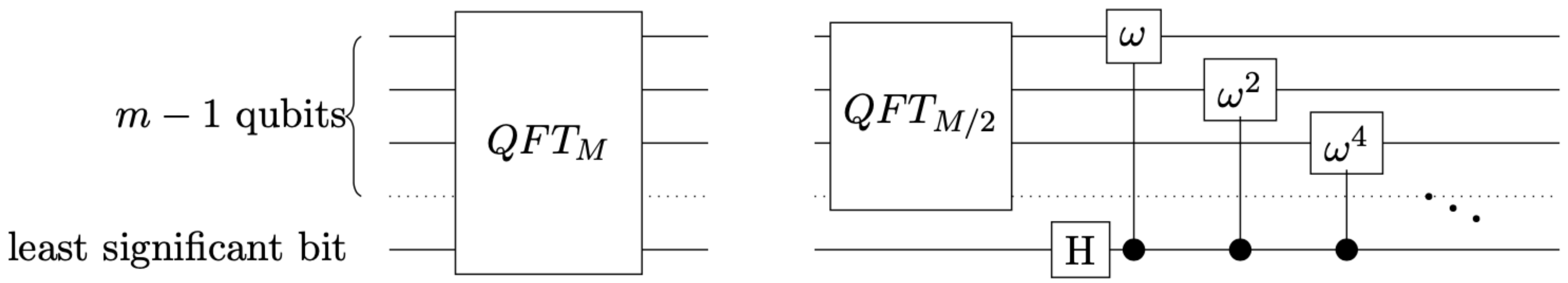


Figure 5.5: QFT_M is reduced to $QFT_{M/2}$ and M additional gates

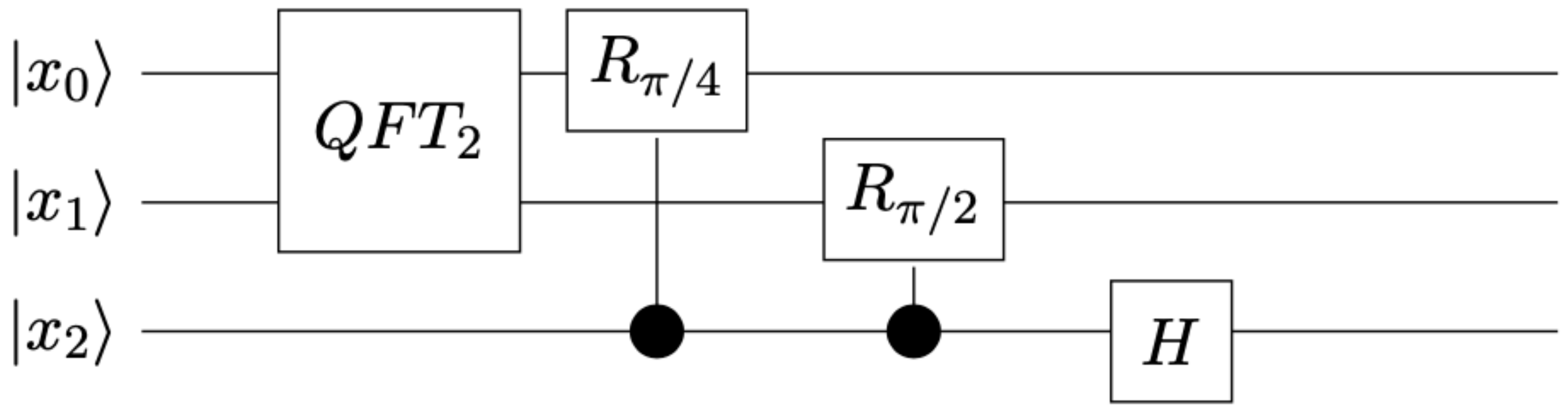


Figure 5.6: First Iteration

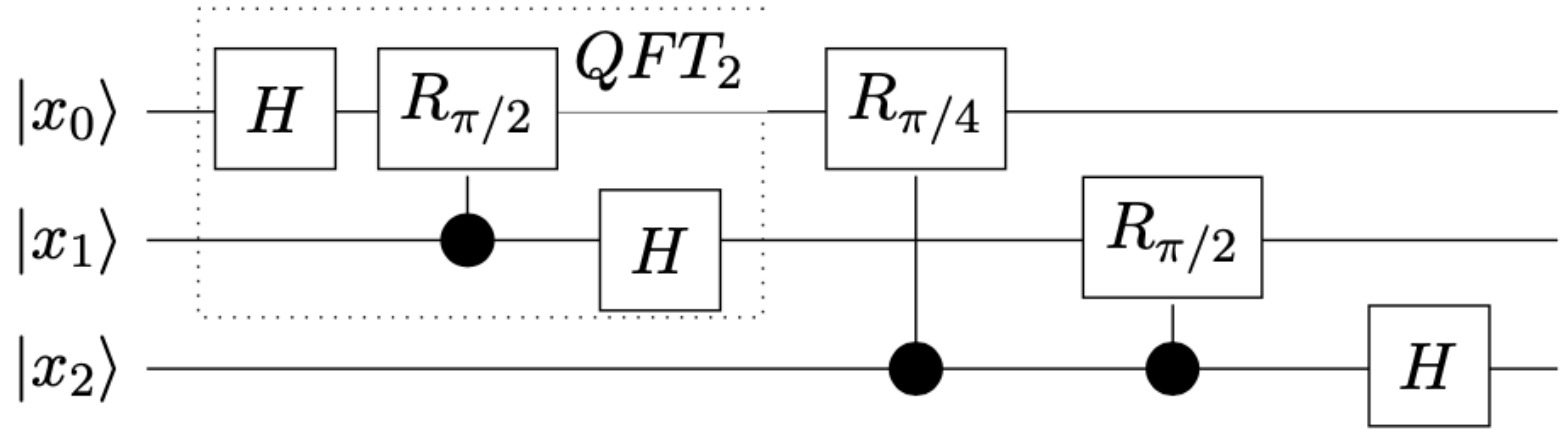
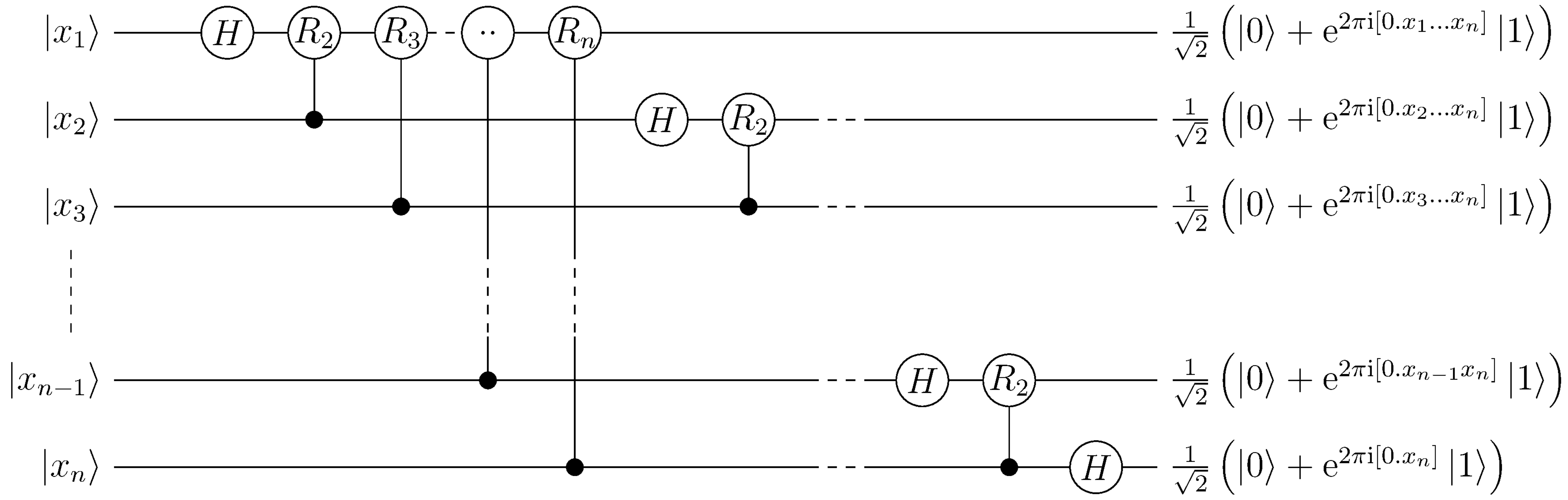


Figure 5.7: Second Iteration. Recall that $H = QFT_1$

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



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

Can we build a scalable Quantum Computer?

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

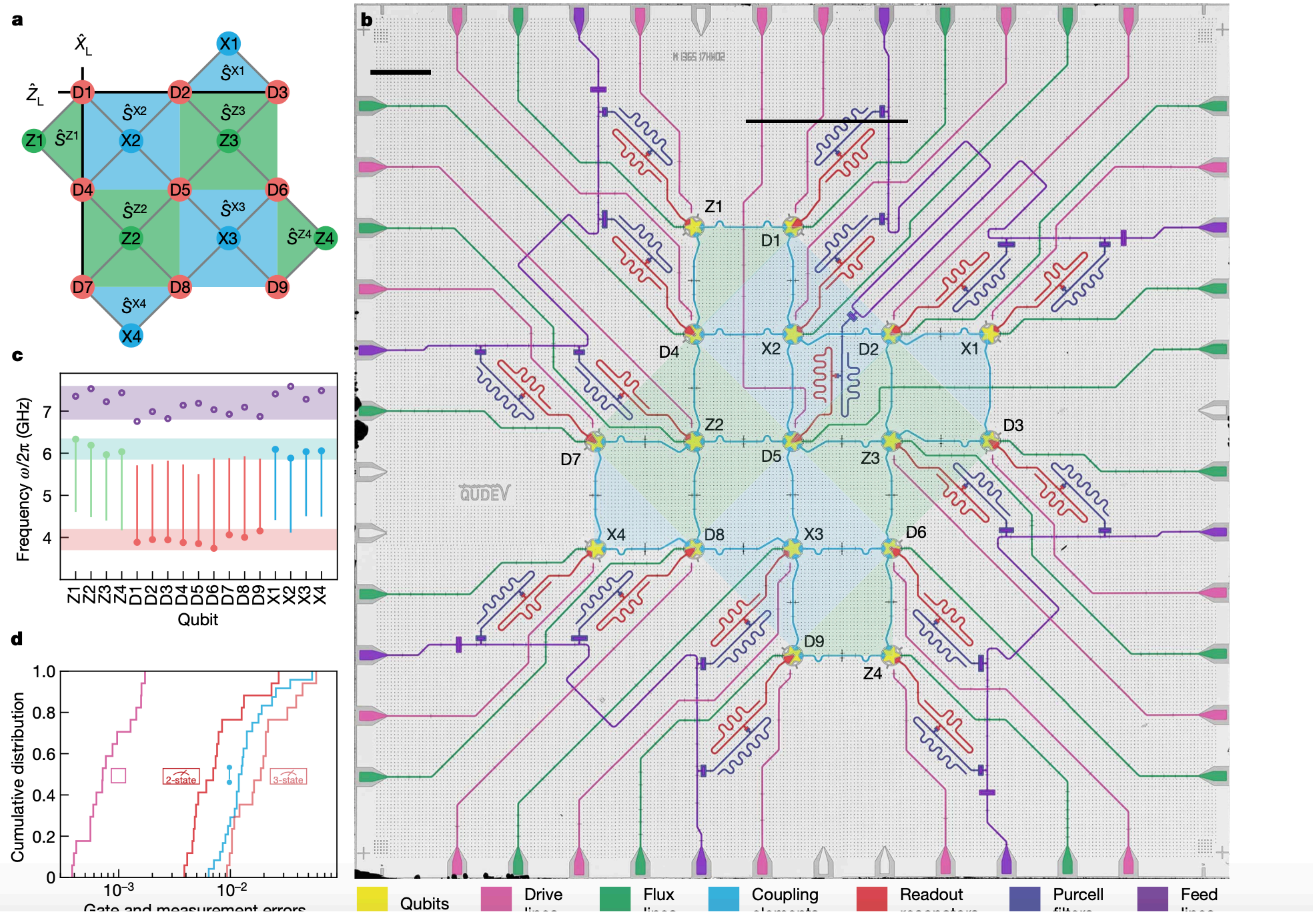
- Fighting quantum decoherence with entanglement
- Quantum Error Correction (QEC)

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

Fault-Tolerant Quantum Memory

Nature 627 (2024) 778

Article



Fault-Tolerant Universal Quantum Gate

Nature 605 (2022) 675

Article

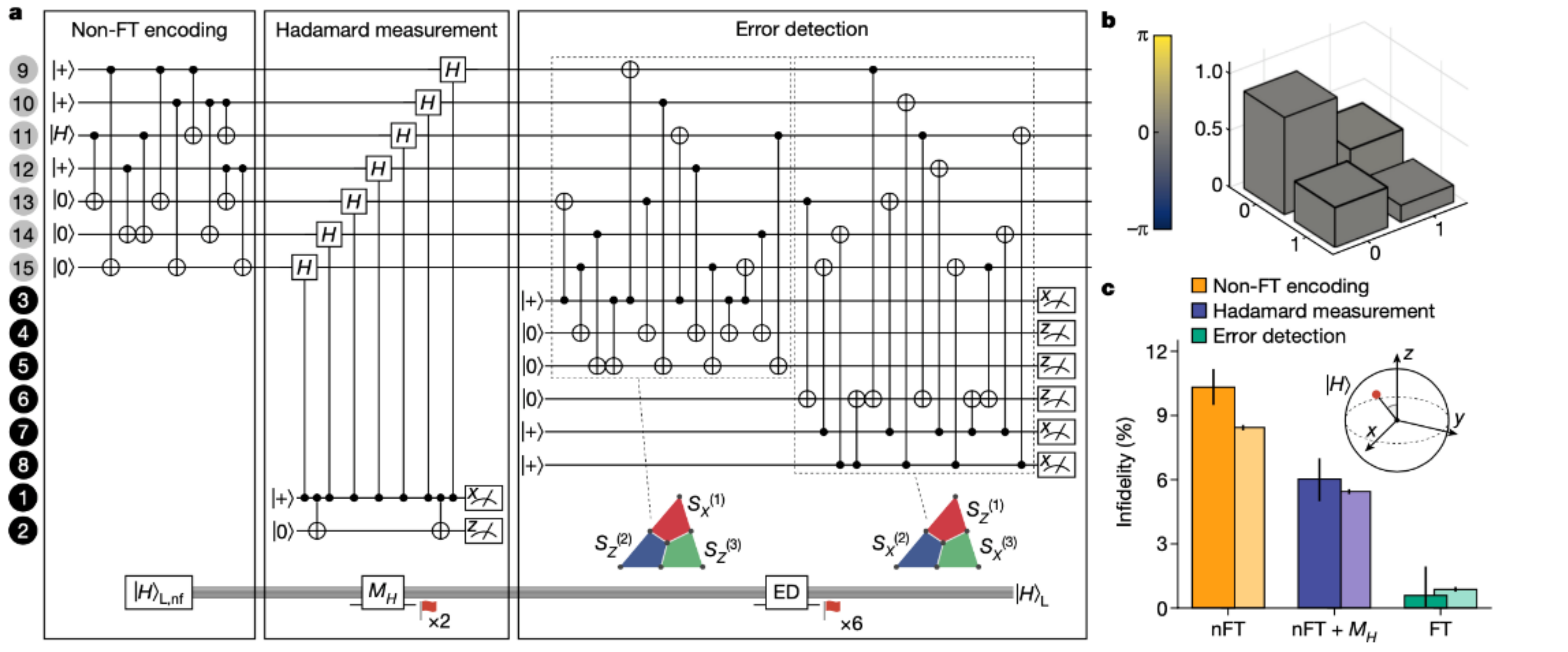


Fig. 4 | Fault-tolerant generation of a logical magic state $|H\rangle_L$. a, The magic state is prepared non-fault-tolerantly in a first step, where a physical magic state $|H\rangle$ is mapped to the logical state $|H\rangle_{L,nf}$ encoded in the data qubits at positions 9 to 15 in the ion string (see labels at left of circuit). Thereafter, a FT measurement of the Hadamard operator (M_H) is carried out. Two auxiliary qubits herald that the prepared state is a +1 eigenstate of the Hadamard operator but also that no dangerous error occurred during the measurement. The magic-state preparation is concluded with an error-detection (ED) block that measures the three X-type and Z-type stabilizers each in an FT fashion. The first part of the error-detection circuit (first dashed box), measures $S_X^{(1)}$, $S_Z^{(2)}$ and $S_Z^{(3)}$, whereas the second part measures $S_Z^{(1)}$, $S_X^{(2)}$ and $S_X^{(3)}$. The magic-state preparation is discarded and repeated in case of a non-trivial syndrome of the eight auxiliary qubits 1 to 8. **b**, Logical state tomography (see 'Transversal fault-tolerant operations') after FT magic-state preparation. The phase of the complex amplitudes is encoded in the colour of the three-dimensional bar plot and the wireframes depict ideal results. Phase deviations from the ideal density matrix are smaller than 50 mrad whereas amplitude deviations are smaller than 0.007. **c**, The decrease in infidelity of the logical magic state (red marker on Bloch sphere) after each step of the FT preparation procedure is observed experimentally and captured by depolarizing noise simulations (experimental and simulation results depicted darker and lighter, respectively).

Quantum error and statistical model

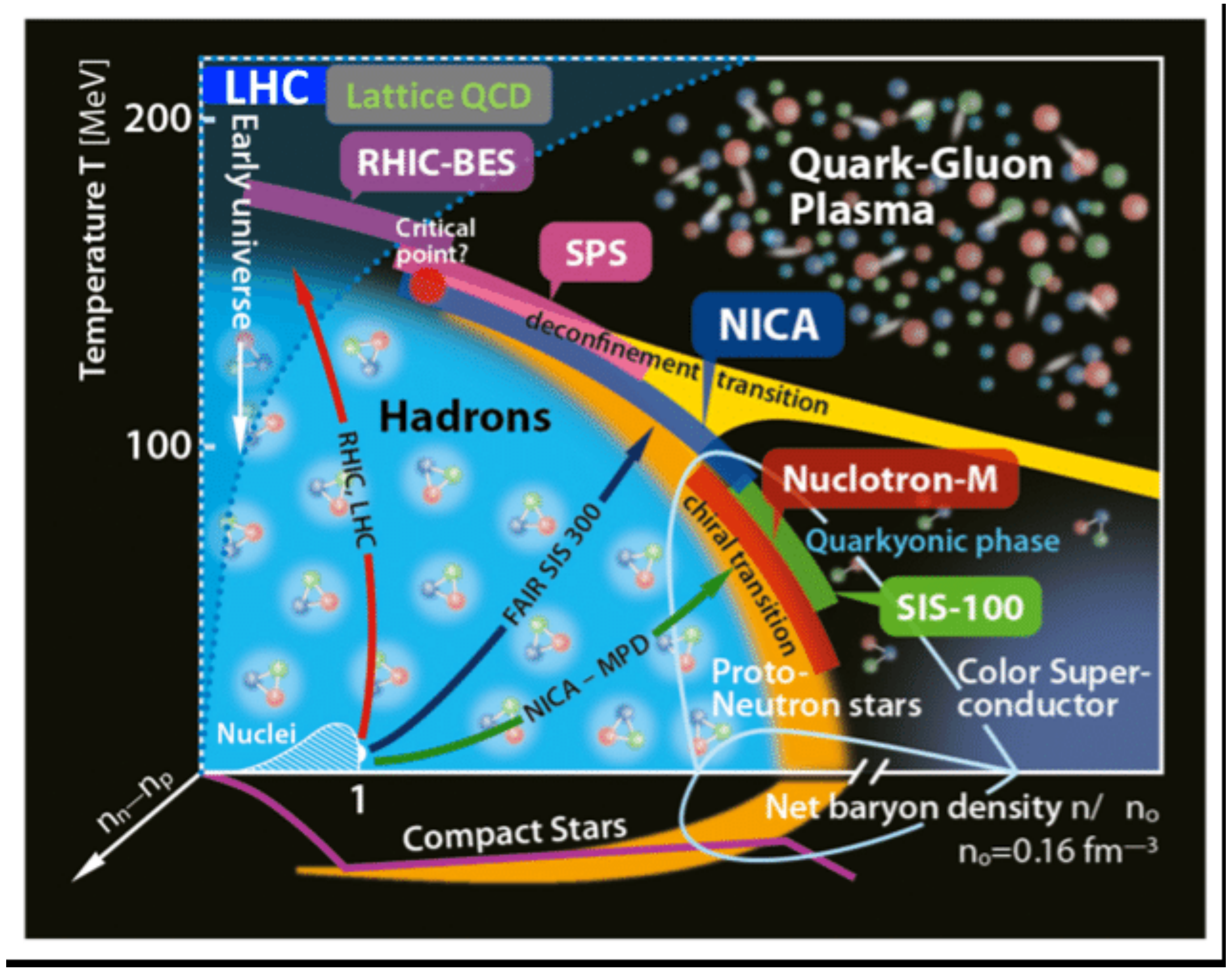
- Specific quantum code
- Modeling quantum error pattern
- Mapping quantum error pattern to statistical physics model
- cf. simple case: Dennis et al, J. Math. Phys. 43 (2002) 4452

Quantum Error Detection/Correction

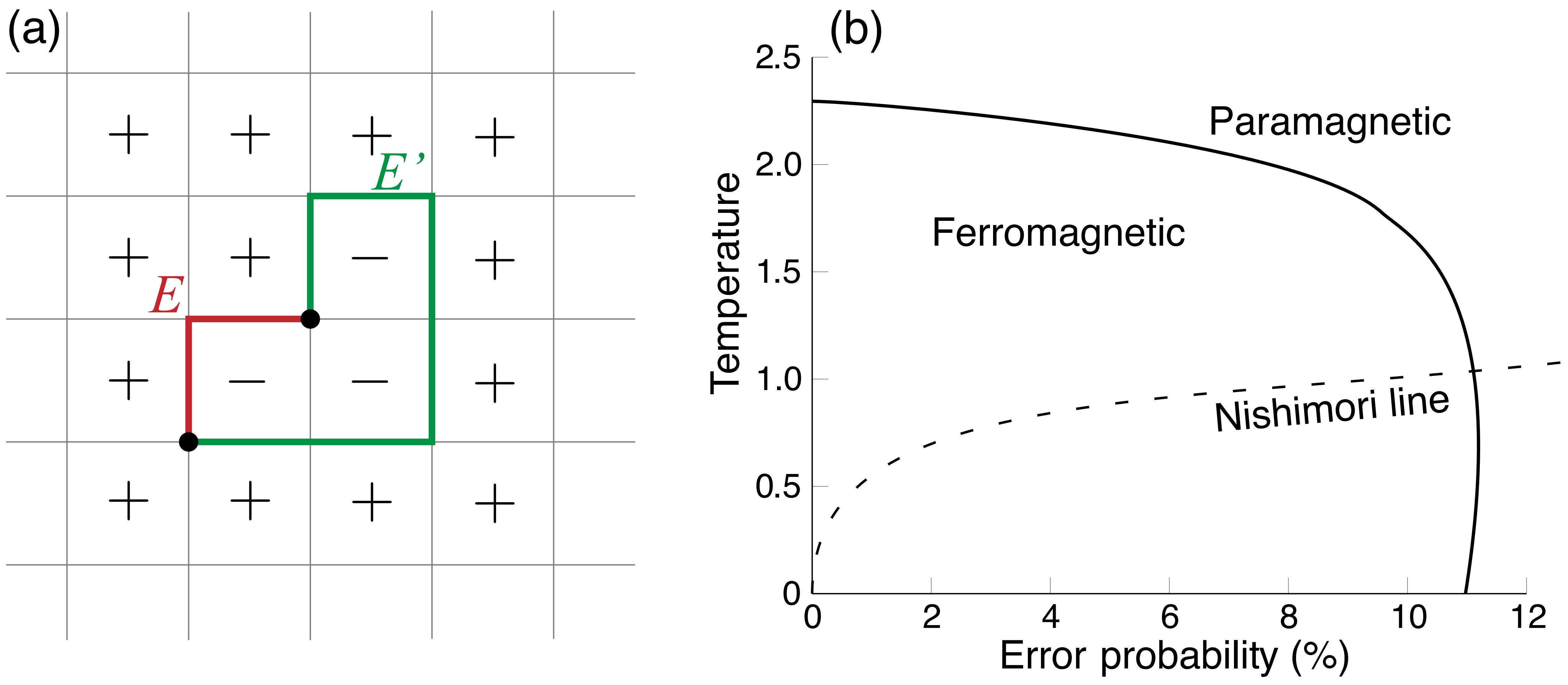
- Check whether error happens via the measurement of "ancilla" qubits: measurement result is called syndrome (quantum error detection)
- From the syndrome, guess quantum error probabilistically
- Correct quantum error

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 physics model

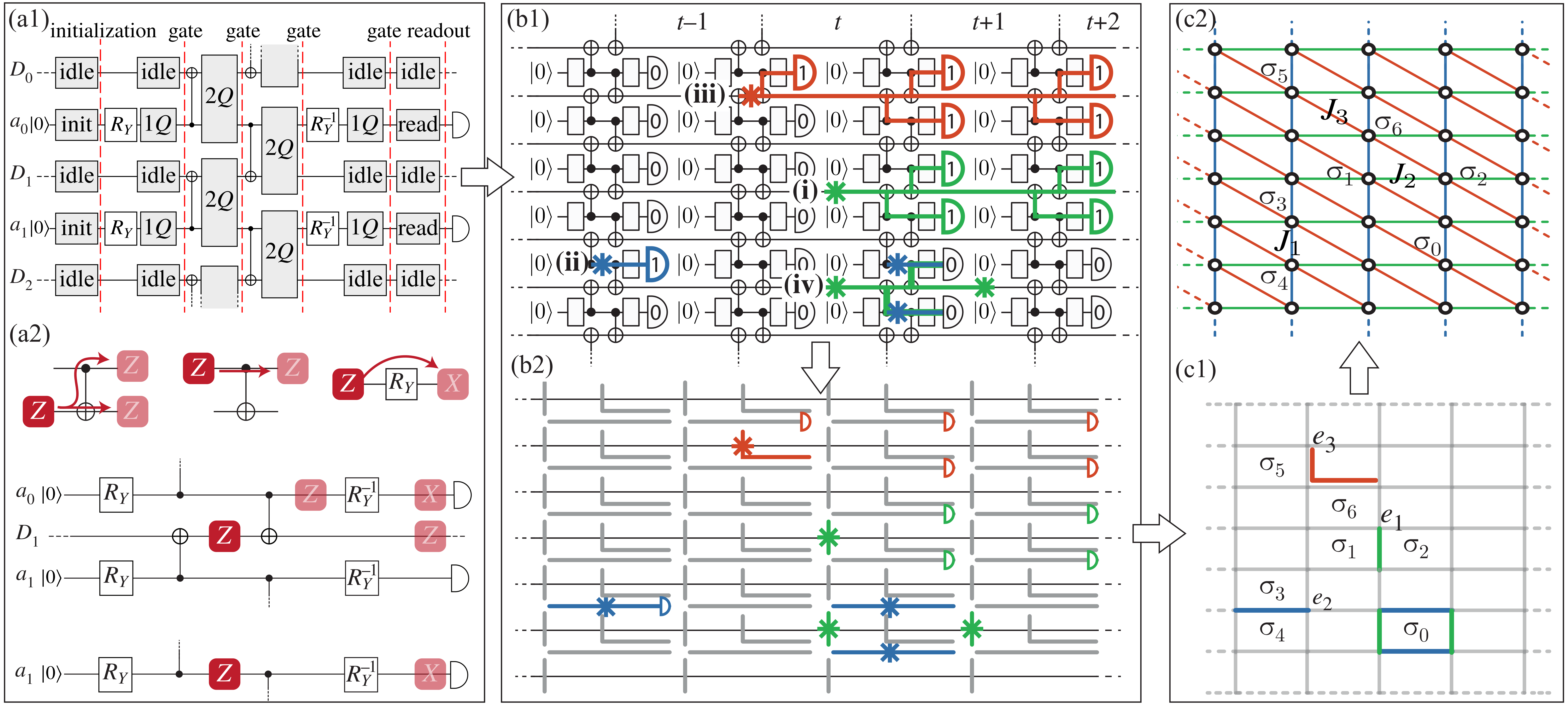


arXiv.2201.00202

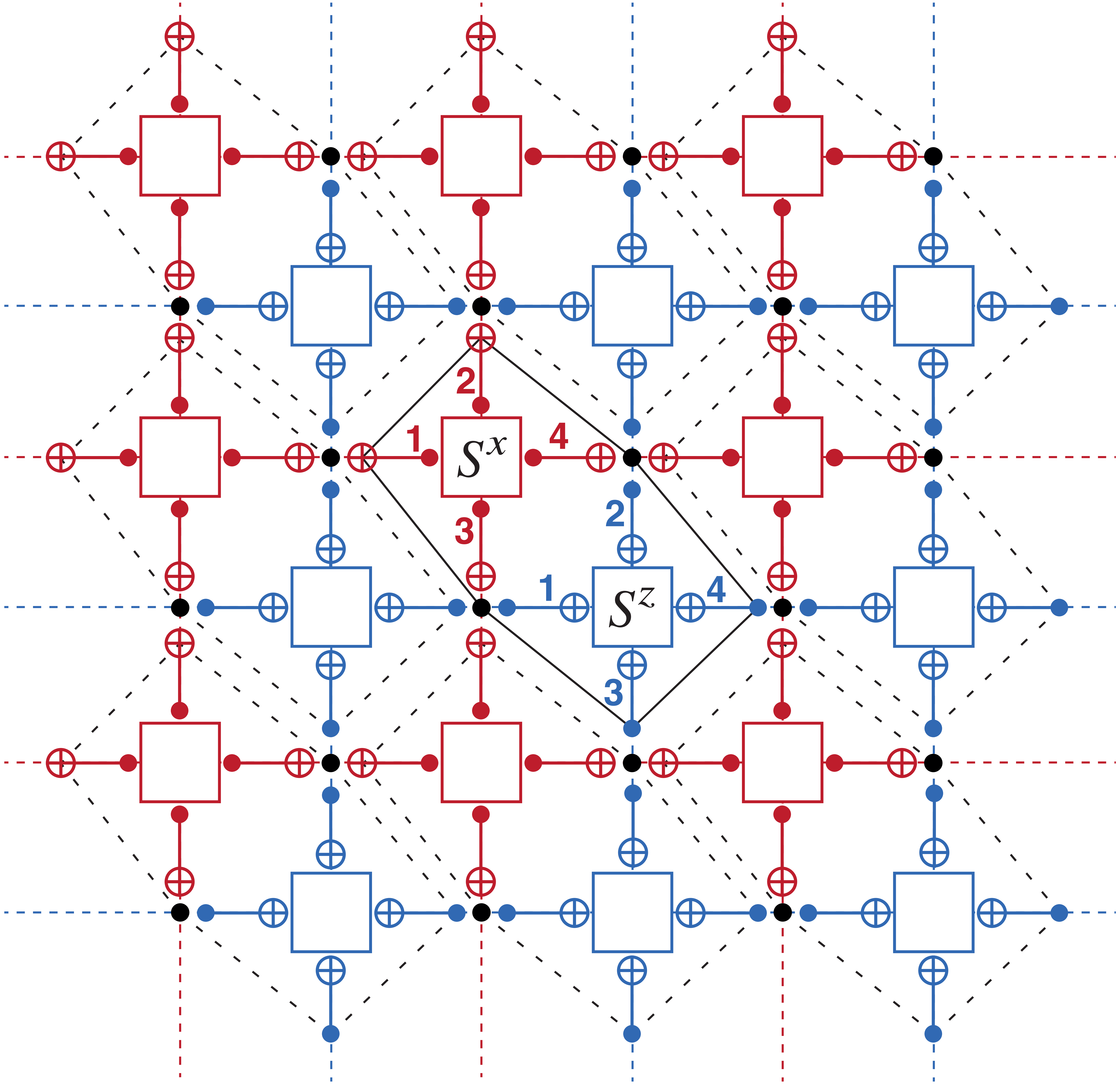


arXiv.2412.14004

- 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), Quantum 6 (2022) 618



arXiv:2412:14004: Toric code circuit



Quantum Error Models and Mapped Statistical physics models in Toric Code

- Random bit flip (σ_x) error or phase flip (σ_z) error

 —〉 2–D Ising model with quenched anti–ferromagnetic coupling

- Random bit flip error or phase flip error + syndrome measurement error

 —〉 3–D $Z(2)$ gauge theory with quenched anti–ferromagnetic coupling

Quantum Error Models and Mapped Statistical physics models in Toric Code

- Independent (σ_x), (σ_z) error + syndrome measurement error

  —> 3-D $Z(2)$ gauge theory

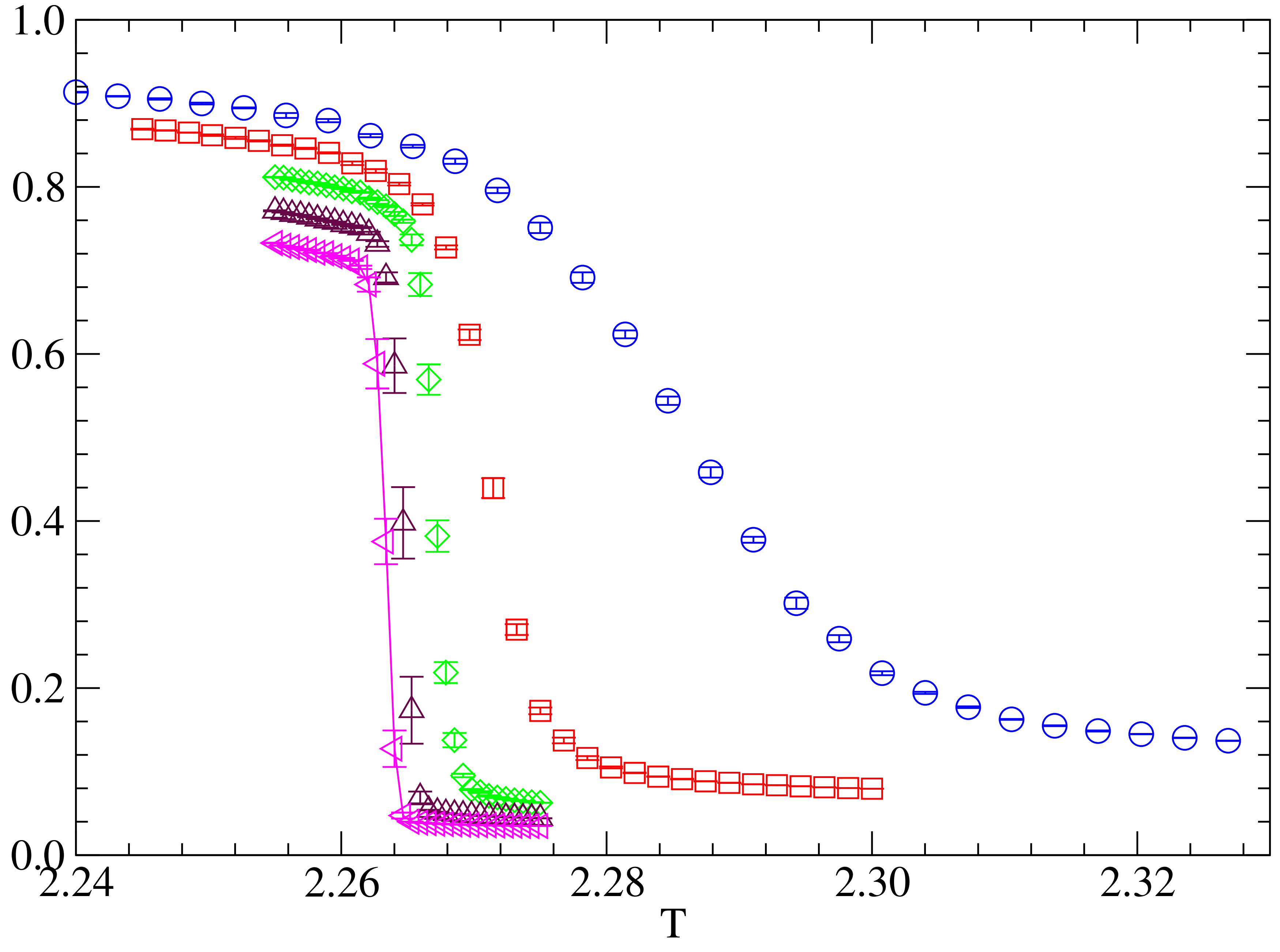
  with quenched anisotropic anti-ferromagnetic coupling

- Depolarizing (i.e., (σ_x),(σ_y),(σ_z)) error + syndrome measurement error

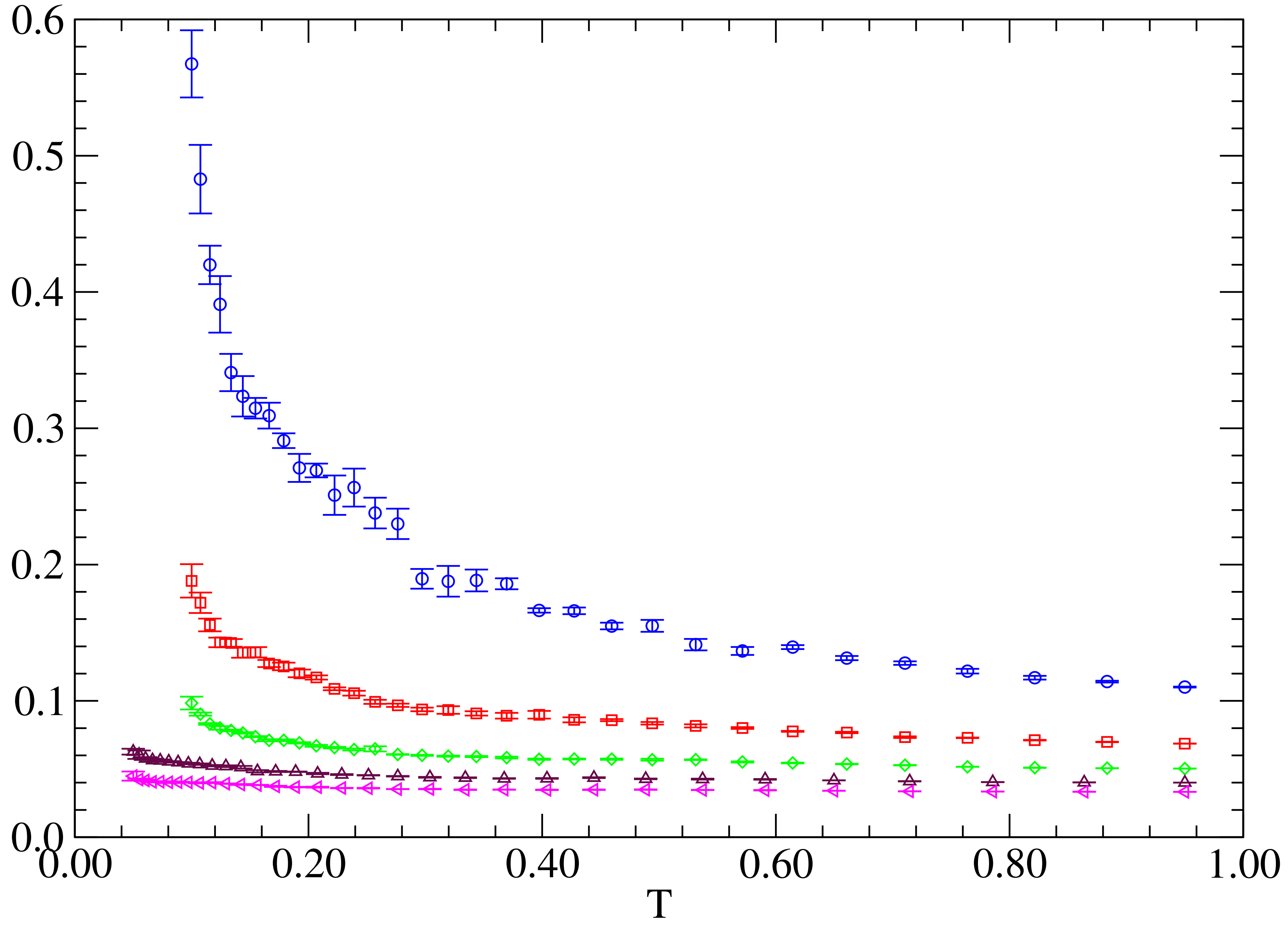
  —> 3-D $Z(2) \times Z(2)$ gauge theory

  with anisotropic quenched anti-ferromagnetic coupling

Polyakov Line, $Z(2) \times Z(2)$ gauge theory

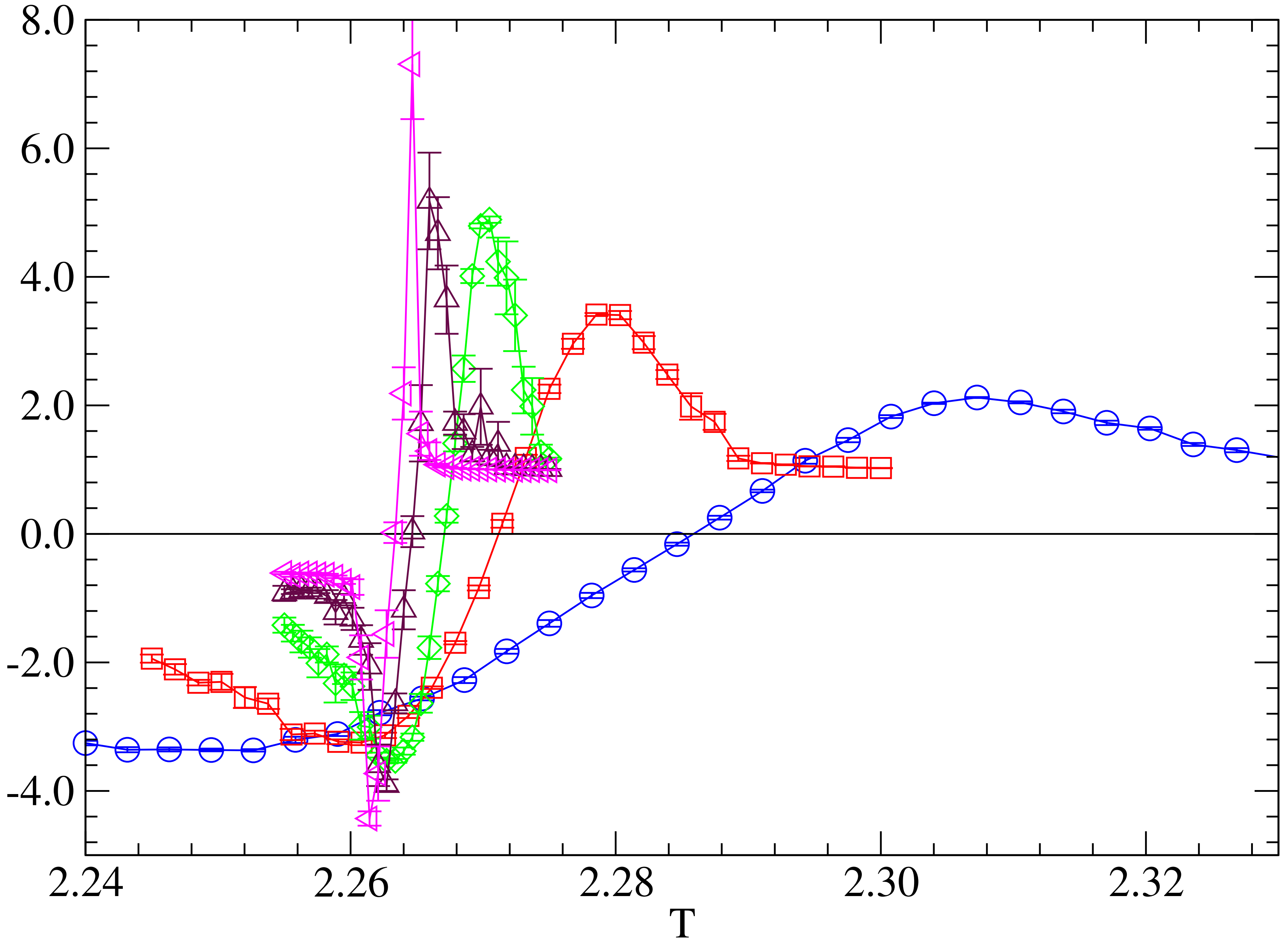


P = 2.88 x 10-5

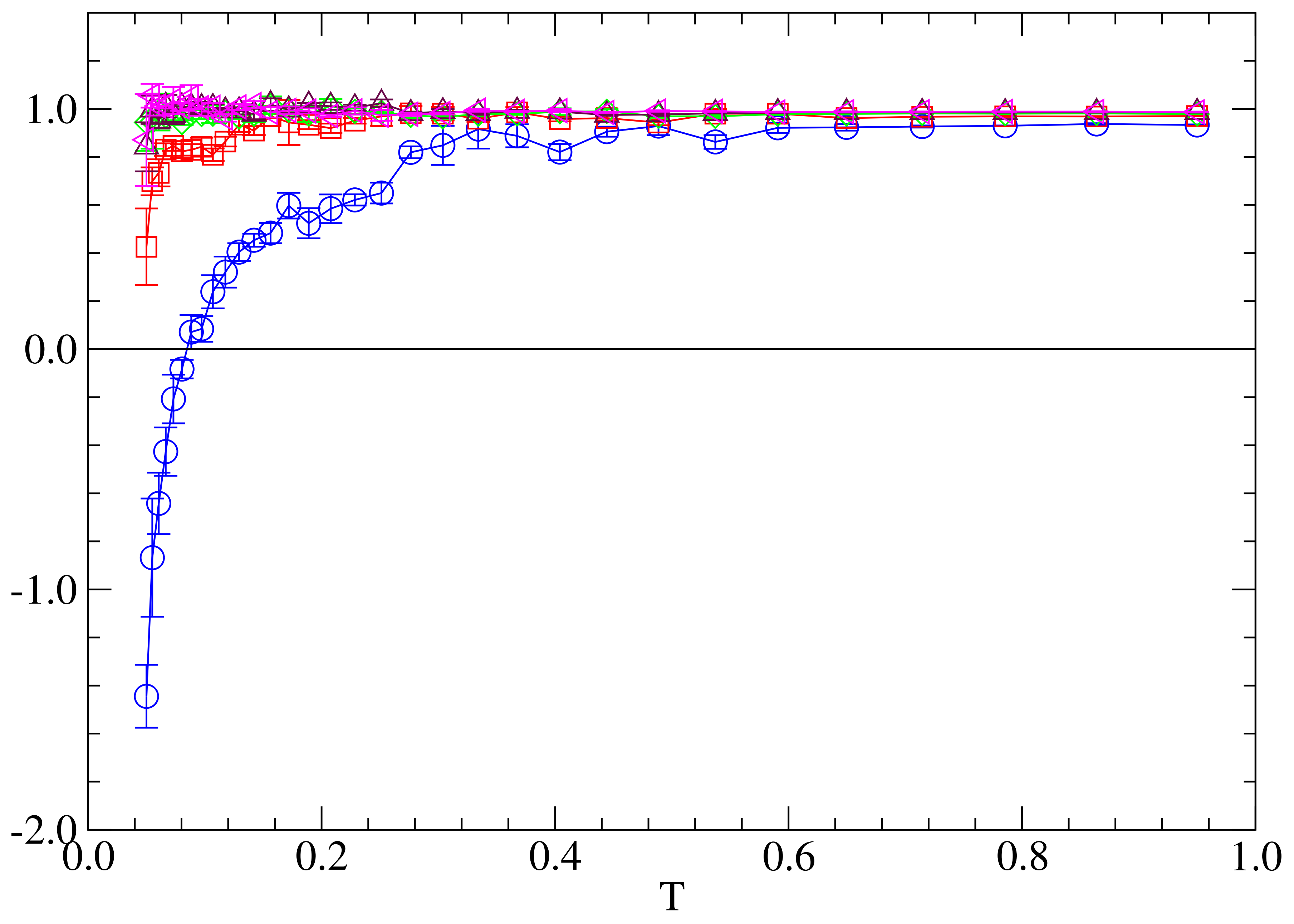


P = 0.0231

Third order cumulant of Polyakov Line, $Z(2) \times Z(2)$ gauge theory

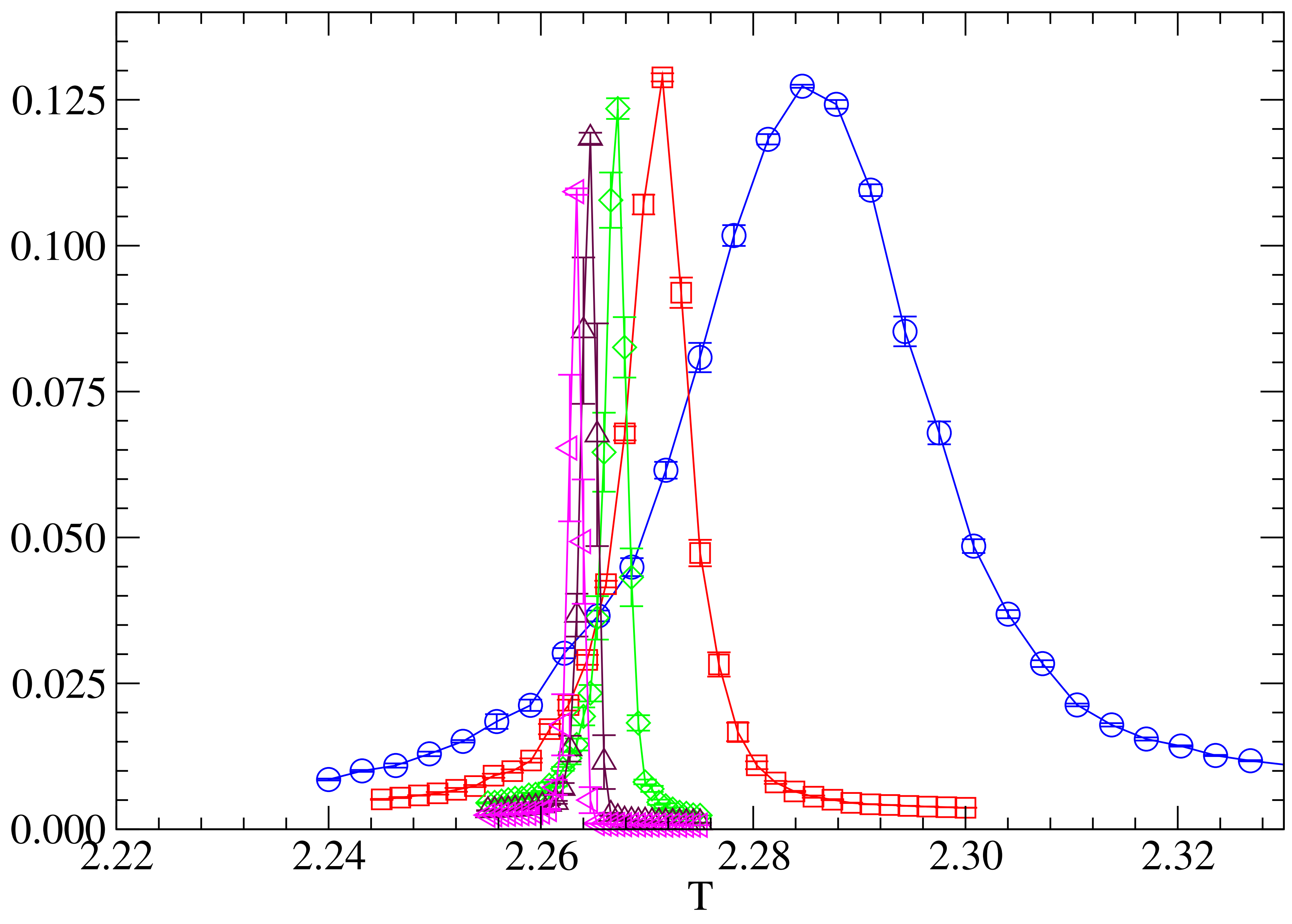


P = 2.88 x 10-5

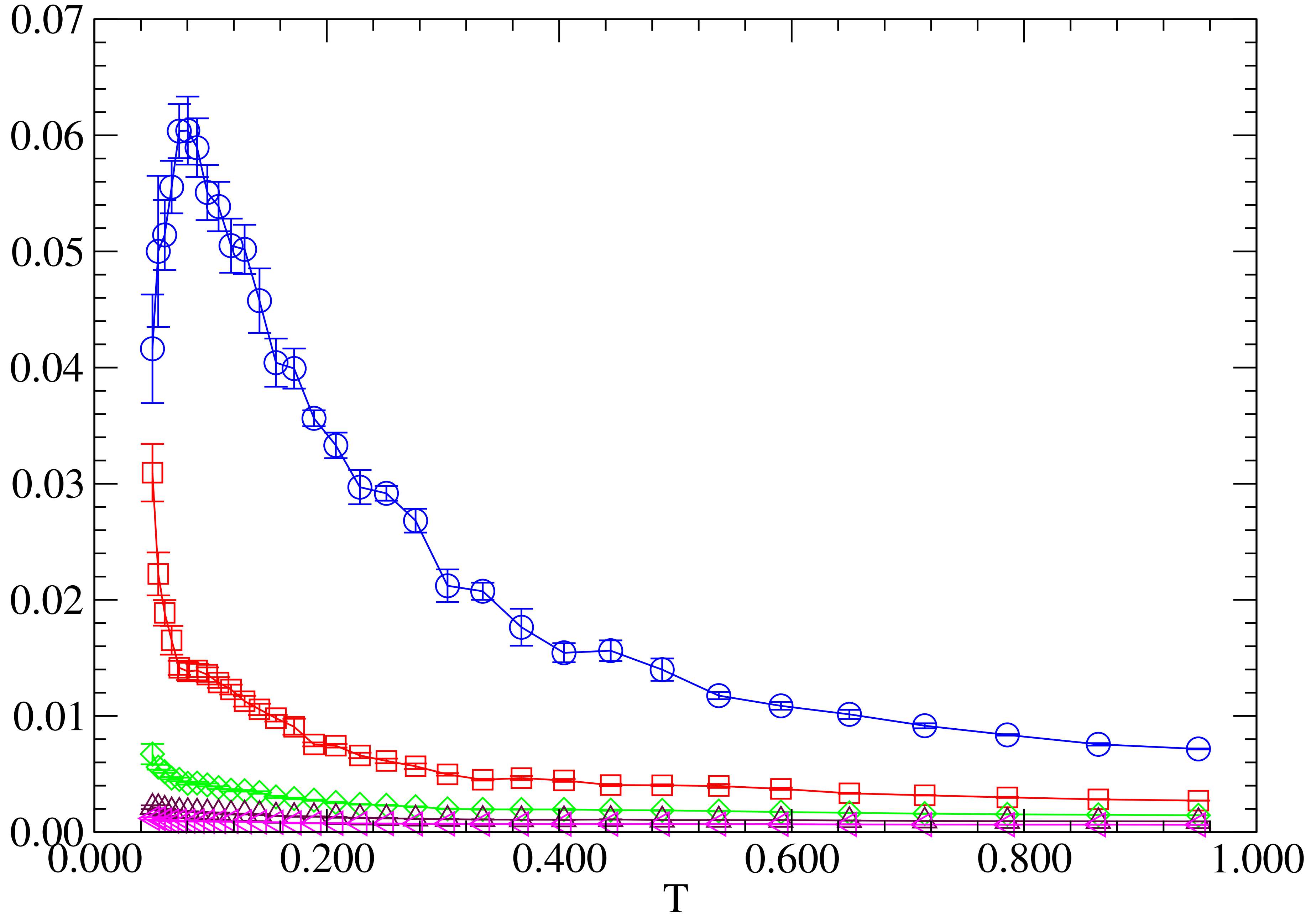


P = 0.0231

Susceptibility of Polyakov Line, $Z(2) \times Z(2)$ gauge theory

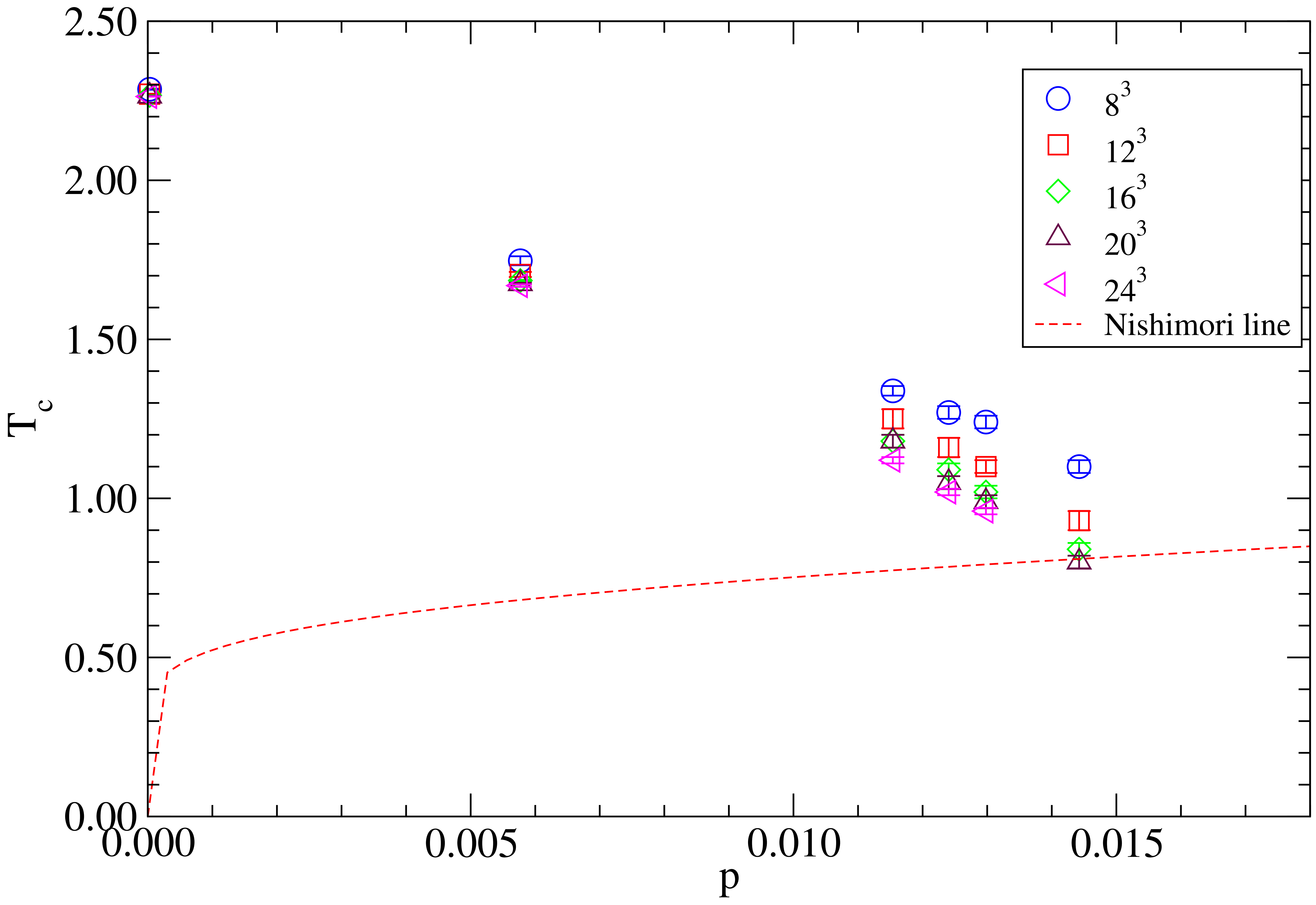


P = 2.88 x 10-5



P = 0.0231

Phase diagram, $Z(2) \times Z(2)$ gauge theory



Conclusion and Discussion

- For some physics problems, Quantum Computer may be a possible way to make a progress
- Quantum Error Correction is crucial for the actual construction of Quantum Computer
- Threshold error probability for the viability of Quantum Error Correction can be studied by parallel tempering MC simulation of quenched statistical physics model
- Threshold probability from the best error decoding algorithm is smaller than that from statistical physics model -> we need a better error decoding algorithm