Computing Topological Field Theories

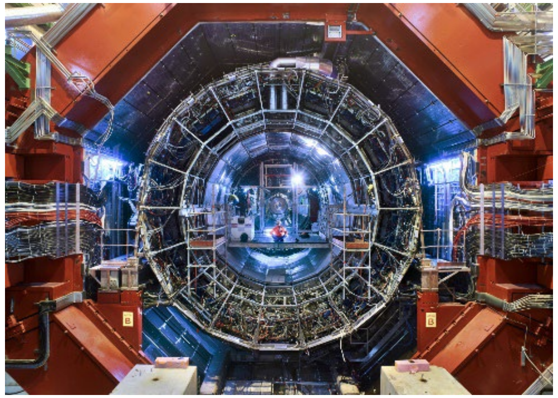
Lecture 3: Tensor Networks & Quantum Computing for Lattice Field Theories

Lena Funcke

School “Recent Advances in Fundamental Physics”, Tbilisi

Overview: computing topological field theories

Numerical computations	Theoretical models	Experiments
Lecture 1: Monte Carlo method High-precision lattice computations Computational issues	**Standard Model** "Real world" Quarks, gluons, Higgs…	**Observables** Spectrum, free energy, entropy, pressure…
Lecture 2: Machine learning Efficient sampling, thermodynamic observables…	**1+1D ϕ^4 theory** Higgs toy model Symmetry breaking…	**LHC, cosmology, …** Heavy-ion collisions, Early-universe physics…
Lecture 3: Tensor networks & quantum computing Topological θ-terms, chemical potentials…	**1+1D Schwinger model** QCD toy model θ-term, confinement…	



(Image credit: ALICE Collaboration / CERN)

Reminder: MCMC is hungry & challenging

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD: ~ 40%**

(lecture on Saturday)

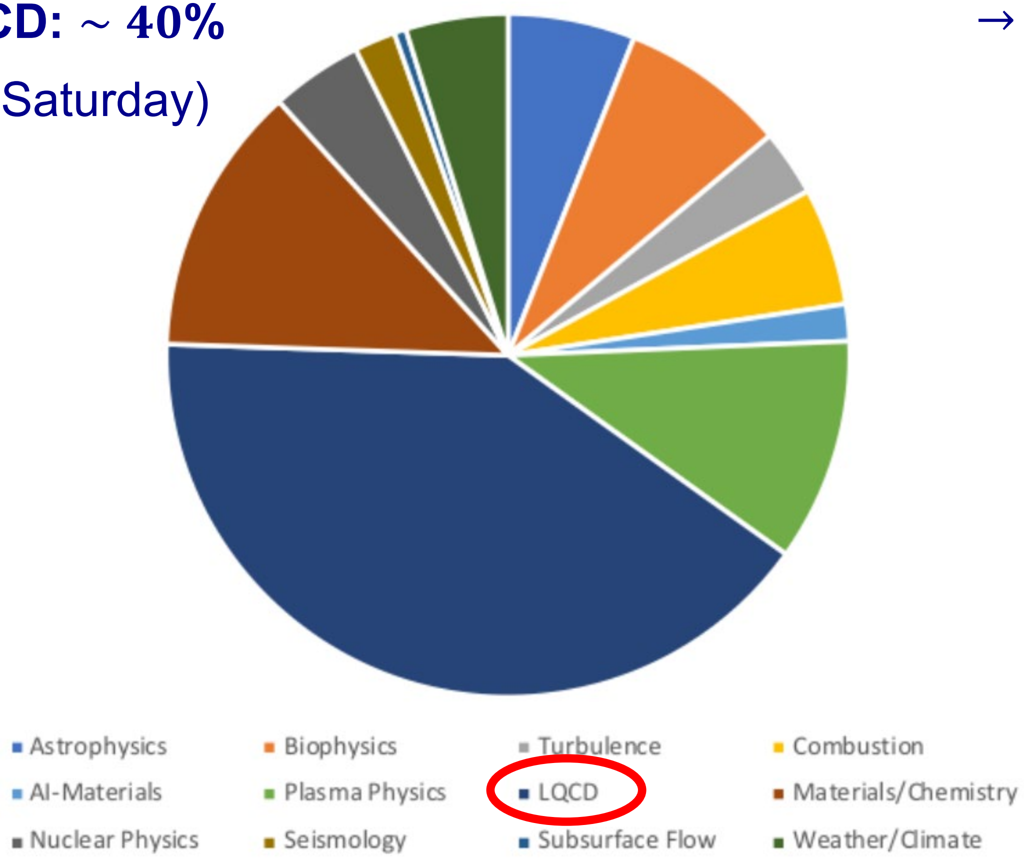
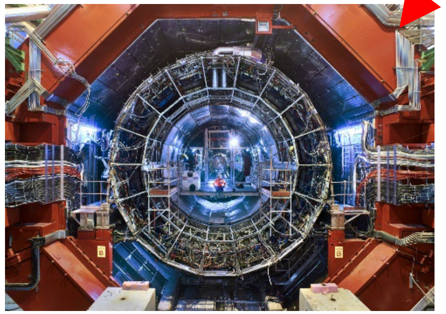


Figure credit: Jack Wells, Kate Clark

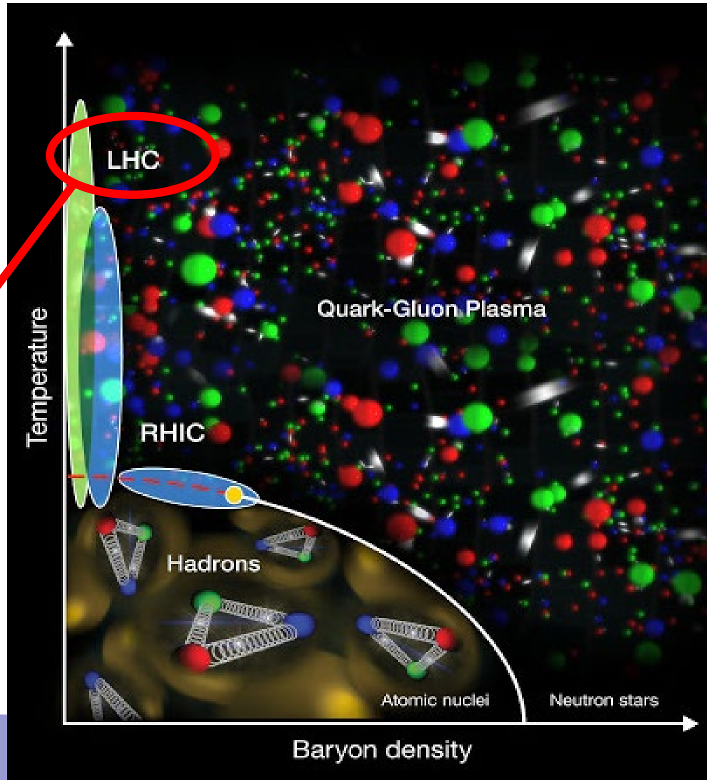
Computational challenges of lattice field theory

No direct computation of thermodynamic observables, …

→ **Machine learning** (lecture yesterday)



(ALICE Experiment / CERN)



Reminder: MCMC is hungry & challenging

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD:** $\sim 40\%$

(lecture on Saturday)

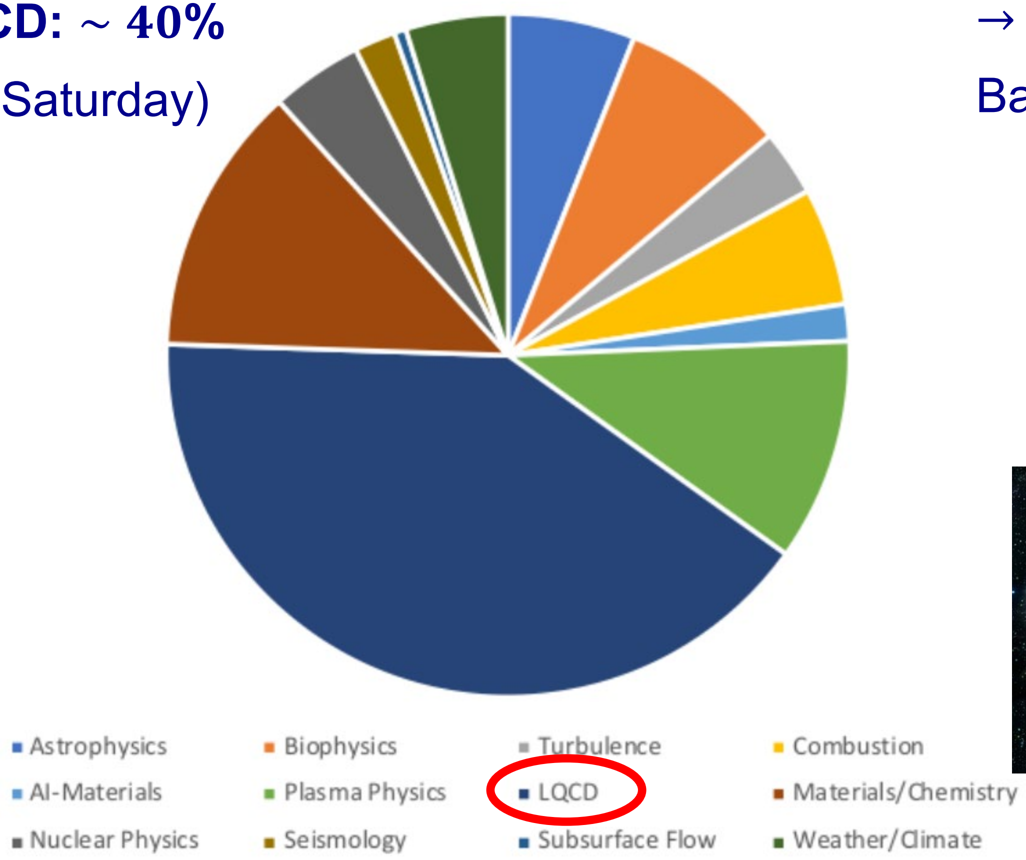


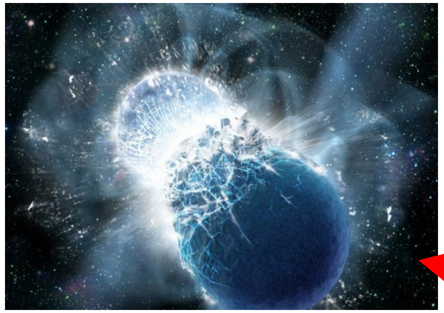
Figure credit: Jack Wells, Kate Clark

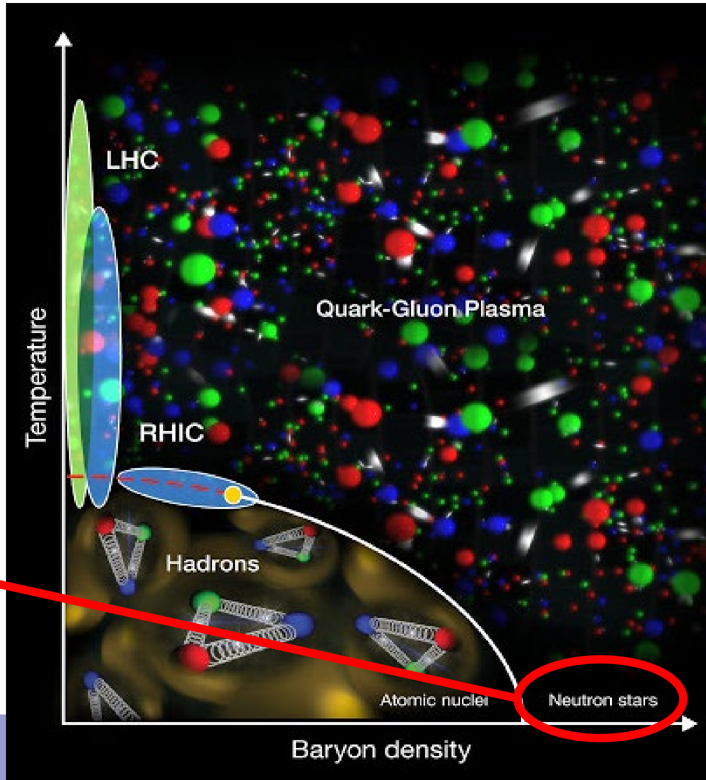
Computational challenges of lattice field theory

No direct computation of thermodynamic observables, …

→ **Machine learning** (lecture yesterday)

Baryon chemical potential, θ-term, real-time evolution, …





Reminder: MCMC is hungry & challenging

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD:** $\sim 40\%$

(lecture on Saturday)

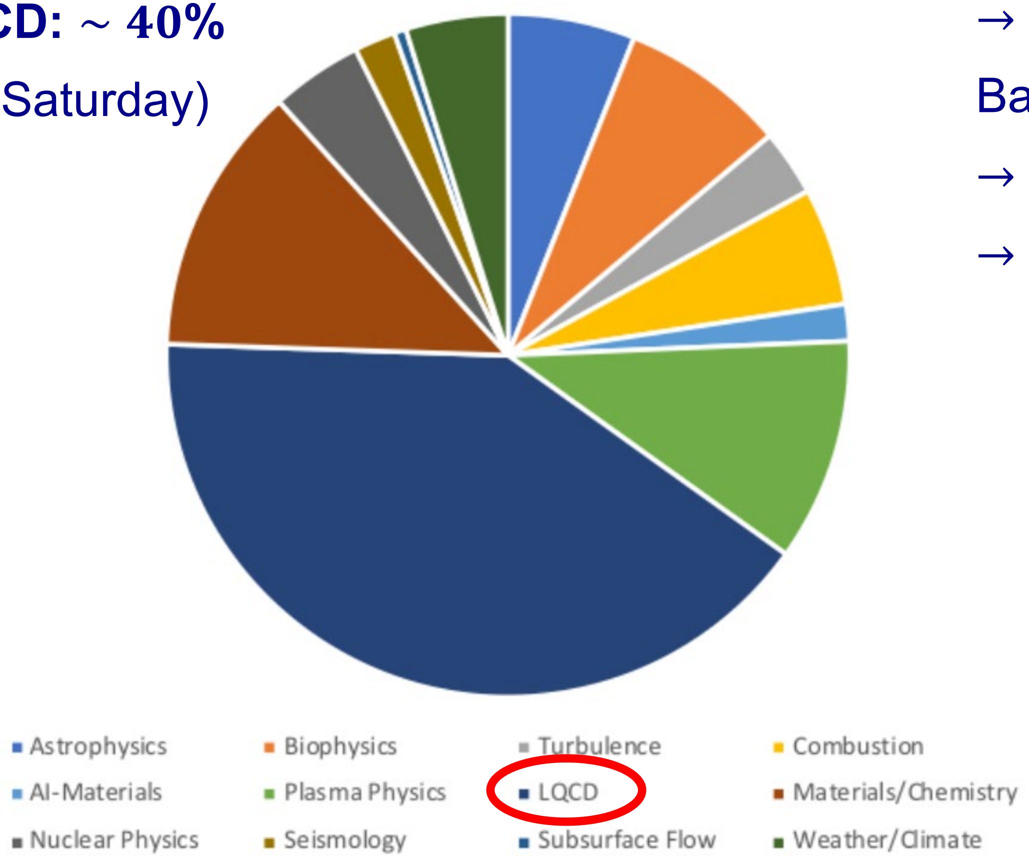


Figure credit: Jack Wells, Kate Clark

Computational challenges of lattice field theory

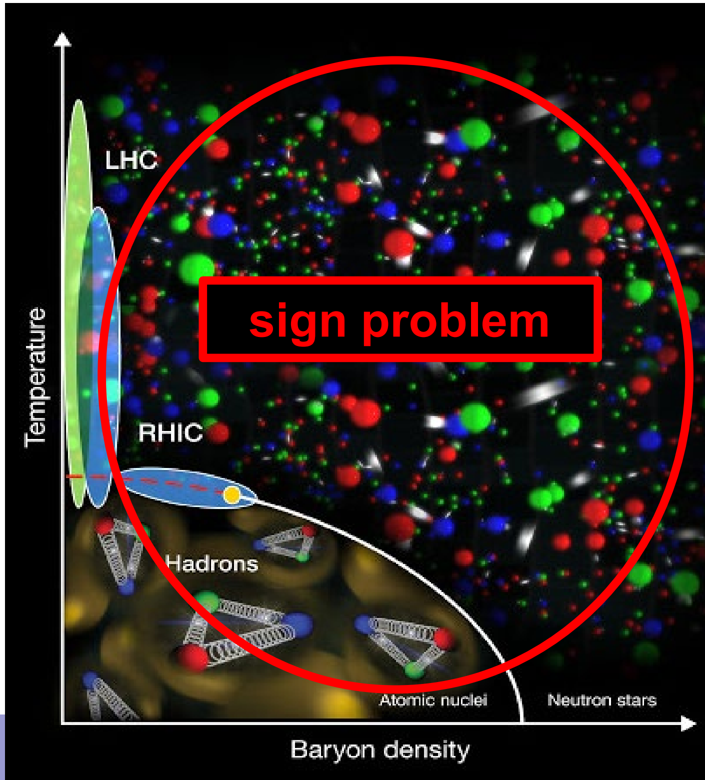
No direct computation of thermodynamic observables, …

→ **Machine learning** (lecture yesterday)

Baryon chemical potential, θ-term, real-time evolution, …

→ **Tensor networks**

→ **Quantum computing (lecture today)**



Reminder: Why is the sign problem exponentially hard?

Example: finite baryon chemical potential

Computing observables

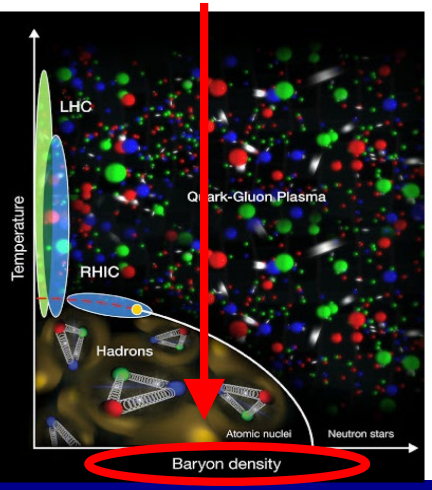
$$\langle O\rangle = \frac{1}{Z}\int DUD\bar{\psi}\,D\psi e^{-S_E} O = \frac{1}{Z}\int DUe^{-S_g} \det M\; O$$

MCMC sampling

Interpretation of $e^{-S_g} \det M$ as *real* probability weight

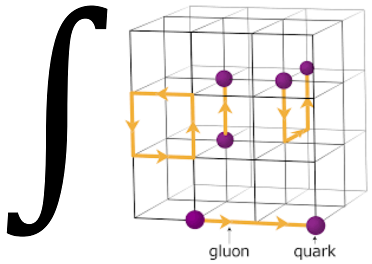
Sign problem

For $\mu \neq 0$, *complex* $\det M$: $[\det M(\mu)]^* = [\det M(-\mu^*)]$



→ no probability weight

→ no MCMC sampling



Reweighting procedure

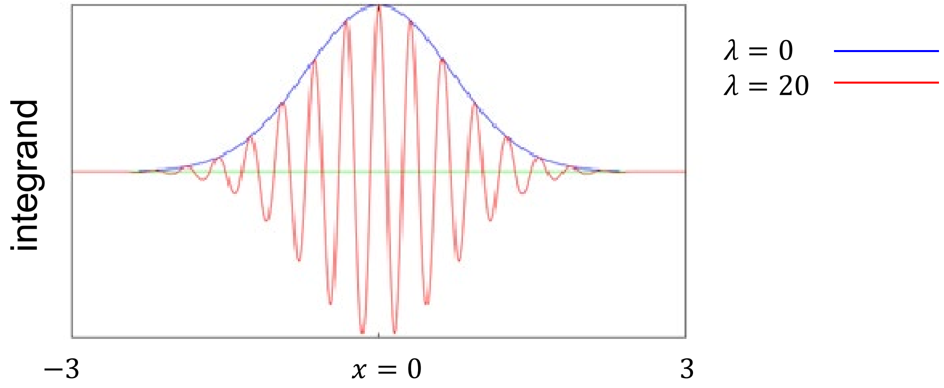
Example: phase quenched theory

$$\langle O\rangle = \frac{\int DUe^{-S_g}\,|\det M|\,e^{i\phi}O}{\int DUe^{-S_g}\,|\det M|\,e^{i\phi}} = \frac{\langle e^{i\phi}O\rangle_{pq}}{\langle e^{i\phi}\rangle_{pq}}$$

Highly oscillatory integrands

Near-cancellation of positive & negative contributions

Sample number grows *exponentially* with volume V



$$\int dx\, \exp(-x^2 + i\lambda x) \to \int dx\, \exp(-x^2)\cos(\lambda x)$$

Do we really need *quantum* computing?

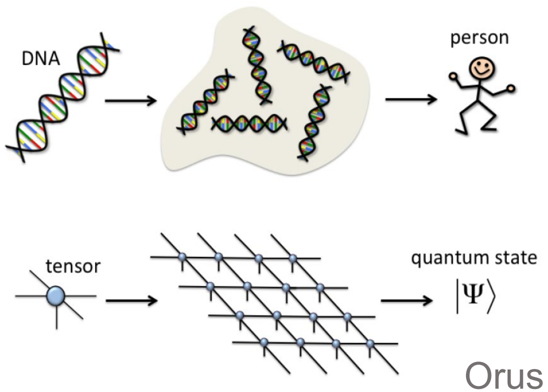
Classical approaches to tackle the sign problem

Tensor networks

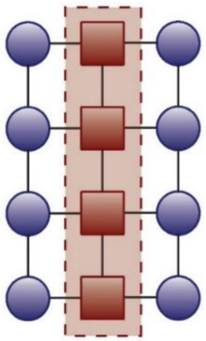
Describe quantum state $|\psi\rangle$ by network of small tensors

E.g. $|\psi\rangle = \sum c_{i_1,\cdots,i_N} |i_1\rangle \otimes \cdots |i_N\rangle = \sum A_1^{i_1} \cdots A_N^{i_N} |i_1\rangle \otimes \cdots |i_N\rangle$

Variational algorithm: minimize energy $E(\vec{\alpha}) = \langle\psi|\mathcal{H}|\psi\rangle$



Orus (2014)



Bañuls et al. (2013)

Other approaches

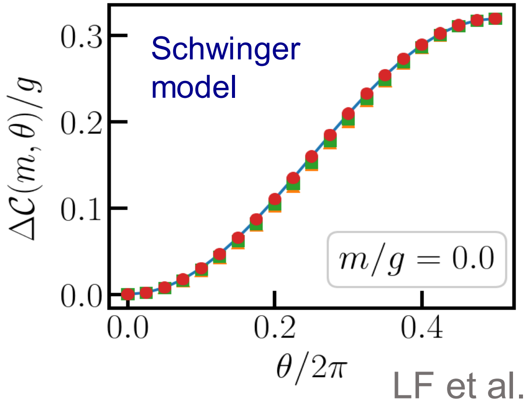
Deep learning for path integral contour deformations,[1] ...

Why is(n't) classical computing enough?

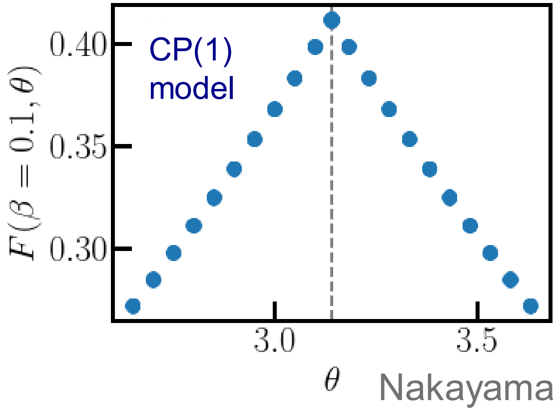
Prospects

Simulate chemical potential, θ-term, real-time dynamics[2]

Mostly focus on 1+1D, first simulations in 2+1D & 3+1D[3]



LF et al. (2020)



Nakayama et al. (2022)

Challenges

No efficient parametrization of highly entangled states

In real-time evolution, tensor size can grow *exponentially*

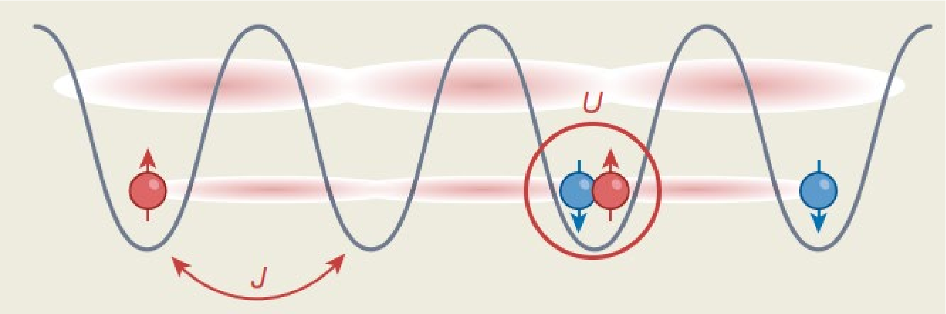
[1] Wynen et al. (2020), ..., [2] Bañuls et al. (2017), Byrnes et al. (2002), Pichler et al. (2016), ...,
[3] Kuramashi et al. (2018), Felser et al. (2020), Akiyama et al. (2019), Magnifico et al. (2021), ...

Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model

Hamiltonian

$$\mathcal{H} = \sum_i -J\left(\hat{a}_i^\dagger \hat{a}_{i+1} + h.c.\right) + \frac{U}{2}\hat{n}_i(\hat{n}_i - 1)$$



Real-time simulation [1]

Analog quantum simulator: ultracold atoms

Classical benchmark: tensor networks (MPS)

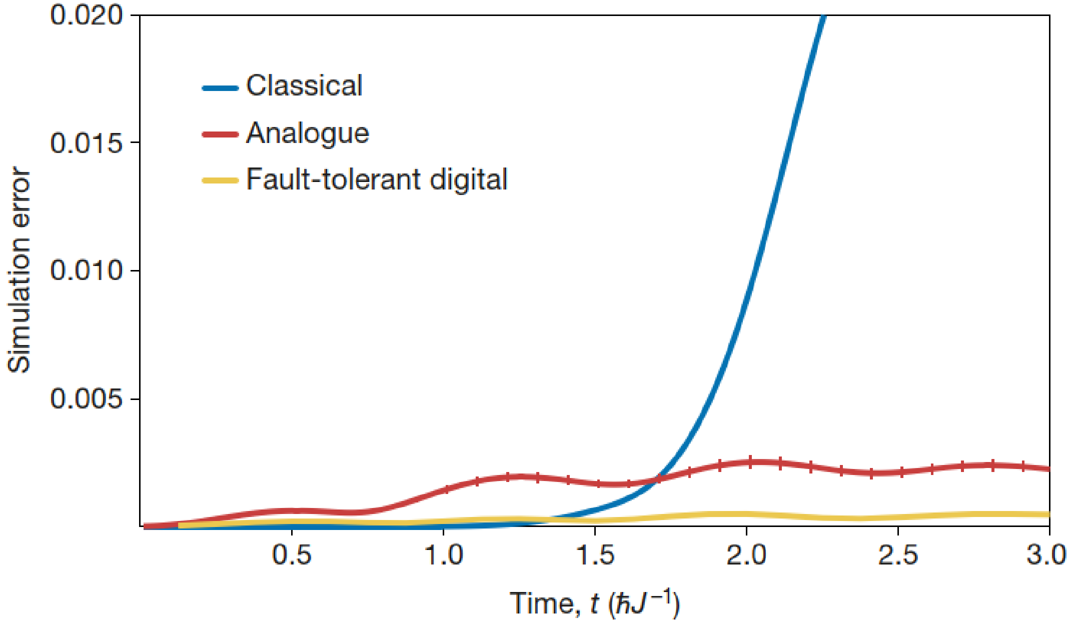
Experimental results

"the controlled [quantum] dynamics runs for longer times than present classical algorithms can keep track of" [1]

Why is(n't) classical computing enough?

Practical quantum advantage in quantum simulation [2]

Here we overview the state of the art and future perspectives for quantum simulation, arguing that a first practical quantum advantage already exists in the case of specialized applications of analogue devices



[1] Trotzky et al. (2012), [2] Daley et al. (2022)

Quantum computing: where do we stand?

Quantum hardware	Quantum algorithms

Achievements

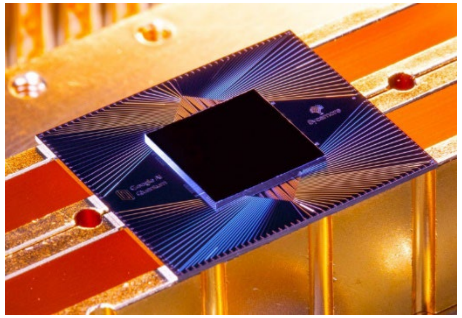
Quantum advantage: outperformed classical computers [1]

Exponential speedup of *specific* classical computations

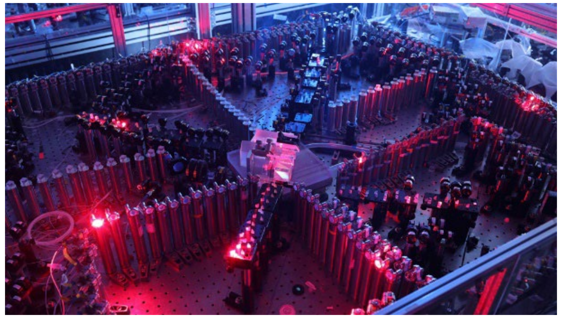
Challenges

$\mathcal{O}(100)$ digital / $\mathcal{O}(1000)$ analog qubits → need more

Noise → need quantum error mitigation / correction



Arute et al. (2019)



Zhong et al. (2020)

[1] Arute et al. (2019), Zhong et al. (2020, 2021), Yulin (2021), Zhu (2021), Madsen (2022).

Applications

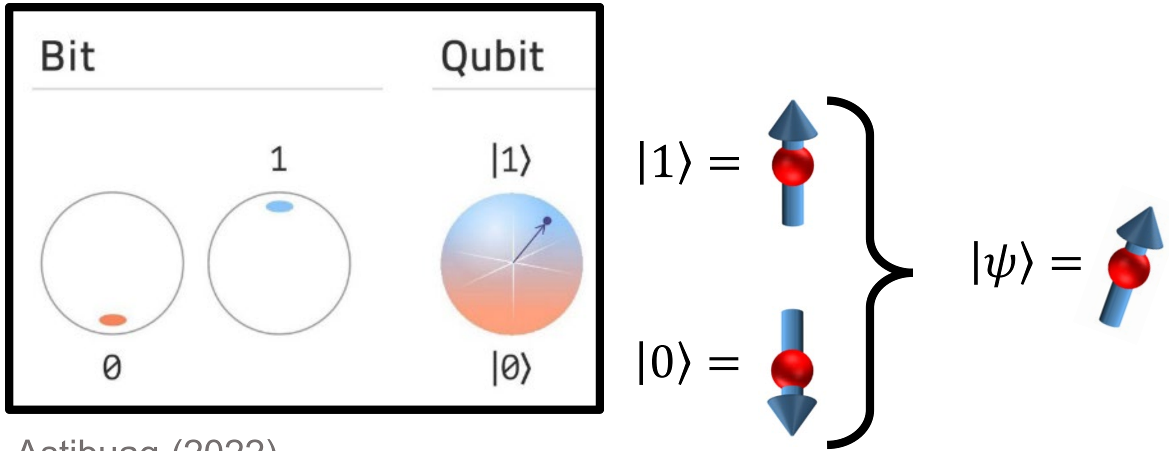
Cryptography, optimization problems, …

Particle / nuclear / condensed matter physics, …

Challenges

New technology → need fundamentally new algorithms

Competition → classical algorithms quickly advance



Astibuag (2022)

Quantum computing: where do we stand?

Example: "quantum advantage" (2019)

Quantum supremacy using a programmable superconducting processor

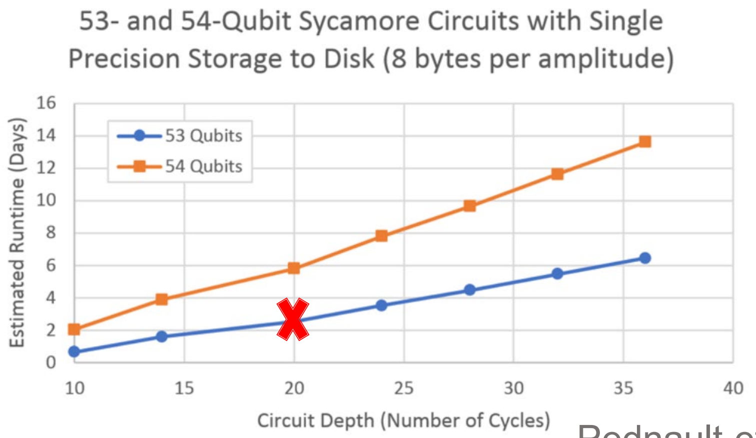
The promise of quantum computers is that certain computational tasks might be executed exponentially faster on a quantum processor than on a classical processor[1]. A fundamental challenge is to build a high-fidelity processor capable of running quantum algorithms in an exponentially large computational space. Here we report the use of a processor with programmable superconducting qubits[2–7] to create quantum states on 53 qubits, corresponding to a computational state-space of dimension 2^{53} (about 10^{16}). Measurements from repeated experiments sample the resulting probability distribution, which we verify using classical simulations. Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years. This dramatic increase in speed compared to all known classical algorithms is an experimental realization of quantum supremacy[8–14] for this specific computational task, heralding a much-anticipated computing paradigm.

→ **Quantum-classical race:**
algorithms and hardware quickly advance
→ **For exponentially hard problems:**
small quantum step ↔ giant classical leap

Example: "classical advantage" (2021)

Closing the "Quantum Supremacy" Gap: Achieving Real-Time Simulation of a Random Quantum Circuit Using a New Sunway Supercomputer

We develop a high-performance tensor-based simulator for random quantum circuits(RQCs) on the new Sunway supercomputer. Our major innovations include: (1) a near-optimal slicing scheme, and a path-optimization strategy that considers both complexity and compute density; (2) a three-level parallelization scheme that scales to about 42 million cores; (3) a fused permutation and multiplication design that improves the compute efficiency for a wide range of tensor contraction scenarios; and (4) a mixed-precision scheme to further improve the performance. Our simulator effectively expands the scope of simulatable RQCs to include the 10×10(qubits)×(1+40+1)(depth) circuit, with a sustained performance of 1.2 Eflops (single-precision), or 4.4 Eflops (mixed-precision)as a new milestone for classical simulation of quantum circuits; and reduces the simulation sampling time of Google Sycamore to 304 seconds, from the previously claimed 10,000 years.



Note: classical runtime improved from 2.5 days (2019) to 304 seconds (2021)

Pednault et al. (2019)

Quantum computing: where will we go?

The Path to Go…

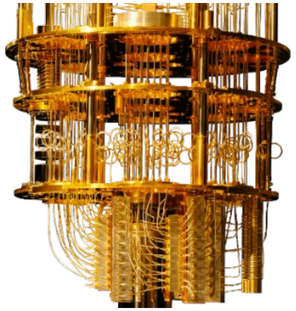
State of the Art

IBM: 27 physical qubits (2019) → 65 (2020) → 127 (2021)

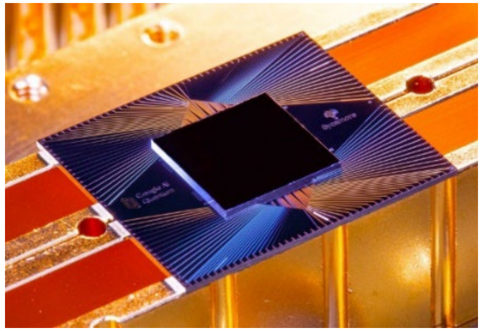
Near Future

IBM: 433 (2022) → 1121 (2023) → 4158 (2025) → …

Google: 1,000,000 physical / 1000 logical qubits (2029)?



Carlow (2018)



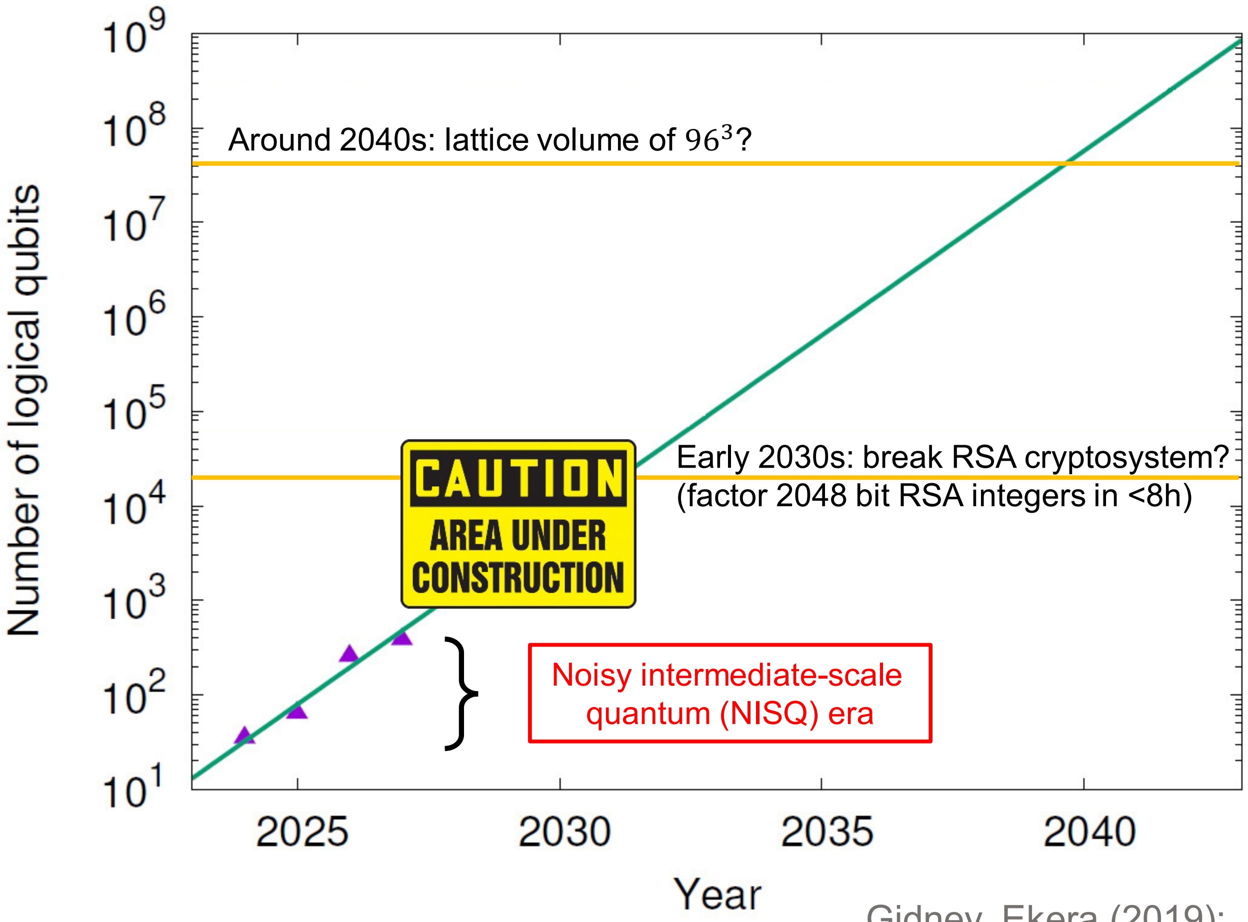
Arute et al. (2019)

Far Future

Need $\mathcal{O}(10^7 - 10^8)$ logical qubits for lattice volume of 96^3

→ Analogy: lattice QCD from 1980s to 2020s?

A Rough Sketch…

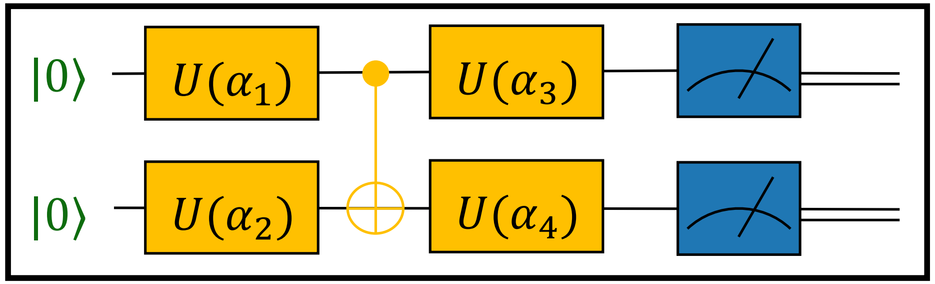


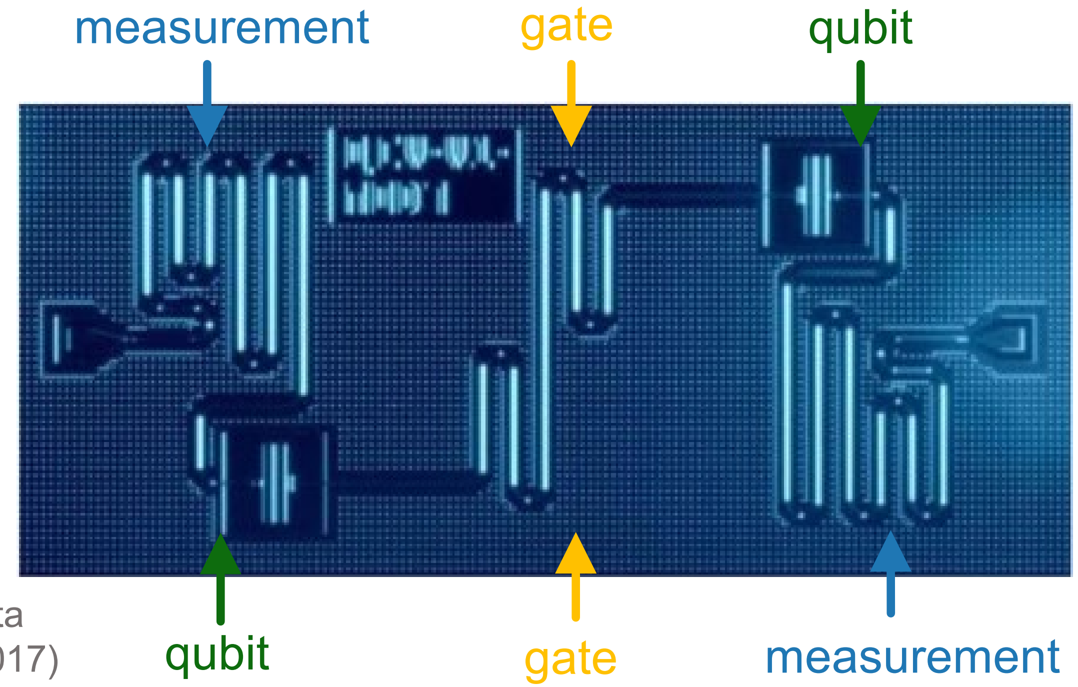
Gidney, Ekera (2019); Kan, Nam (2021)

How can we reduce the noise?

Noisy quantum circuit

$|\psi(\vec{\alpha})\rangle =$





Gambetta et al. (2017)

Error mitigation versus error correction

Problem

Quantum noise: affecting qubits, gates, measurement

Near-term solution

Error mitigation: reduce noise on NISQ devices

Long-term solution

Error correction (EC): fault-tolerant quantum computation

E.g. bit-flip code,[1] Shor code,[2] toric code,[3] GKP code,[4] …

Quantum threshold theorem

For EC, need extra qubits and noise below *threshold*[5]

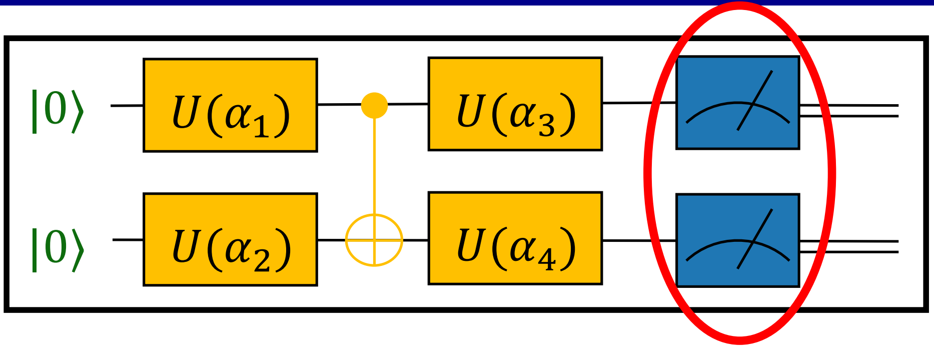
E.g. toric code needs > 1000 extra qubits for $p < 0.1\%$

[1] Peres (1985), [2] Shor (1995), [3] Kitaev (1997), [4] Gottesmann et al. (2001), ...
[5] Shor (1996), Knill et al. (1998), Kitaev (2003), Aharonov et al. (2008)

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

$|\psi(\vec{\alpha})\rangle =$



Readout	Bit Flips	Probability	Noisy Operator
correct	$0 \to 0,\ 1 \to 1$	$(1-p_0)(1-p_1)$	$\tilde{O} = \boldsymbol{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
incorrect … for both outcomes	$0 \to 1,\ 1 \to 0$	$p_0 p_1$	$\tilde{O} = -Z = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
… for outcome 0	$0 \to 1,\ 1 \to 1$	$p_0(1-p_1)$	$\tilde{O} = -\mathbb{I} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$
… for outcome 1	$0 \to 0,\ 1 \to 0$	$(1-p_0)p_1$	$\tilde{O} = \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Operator rescaling method [1]

Goal

mitigate bit-flip errors during readout: $0 \xrightarrow{p_0} 1$ or $1 \xrightarrow{p_1} 0$

Method

replace operators by noisy operators: $\langle\tilde{\psi}|\boldsymbol{O}|\tilde{\psi}\rangle \to \langle\psi|\tilde{O}|\psi\rangle$

Total noisy operator: $\tilde{O}$

$$= (1-p_0)(1-p_1)\boldsymbol{Z} + p_0 p_1(-Z) + p_0(1-p_1)(-\mathbb{I}) + (1-p_0)p_1\mathbb{I}$$

Rescaled (zero-noise) operator:

$$\to \boldsymbol{Z} = \frac{1}{1-p_0-p_1}\tilde{O} - \frac{p_1-p_0}{1-p_0-p_1}\mathbb{I}$$

[1] Single Z operator: Kandala et al. (2017), strings of Z operators: Yeter-Aydeniz et al. (2019), generalizations: LF, et al. (2020), (2021); Alexandrou, et al. (2021a), (2021b)

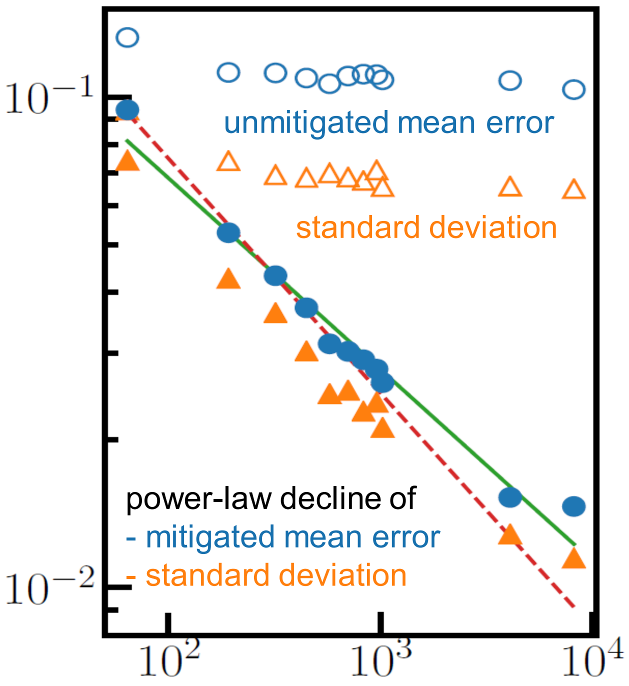
Error mitigation: how can we reduce the noise?

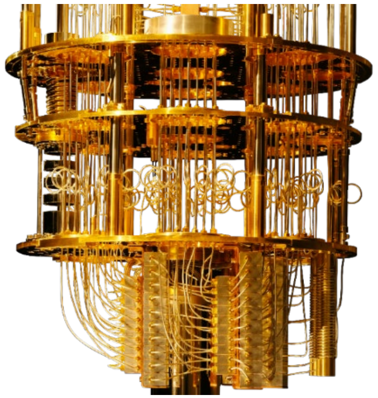
Example: measurement error mitigation

Operator rescaling method[1]

Benchmark: Z and $Z_1 Z_2$ operators on IBM-Q hardware

Result: measurement error reduced by factor 10





LF et al. (2020)

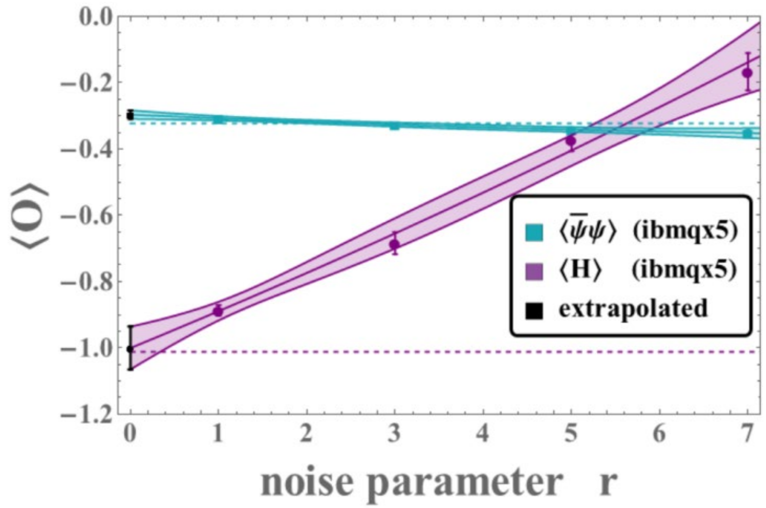
Example: gate error mitigation

Other mitigation techniques

Zero-noise extrapolation,[2] randomized compiling,[3] quasi-probability decomposition,[4] …

Lattice field theory applications

Zero-noise extrapolation for lattice Schwinger model:



Klco et al. (2018)

[1] Kandala et al. (2017), Yeter-Aydeniz et al. (2019), LF et al. (2020); [2] Li et al. (2017), Wallman et al. (2016), [4] Temme et al. (2017), van den Berg (2020), ...

Which field theories have already been simulated?

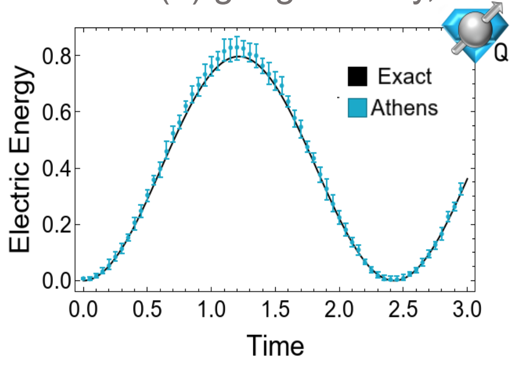
Experimental results on "public" QC

Experimental results on "private" QC

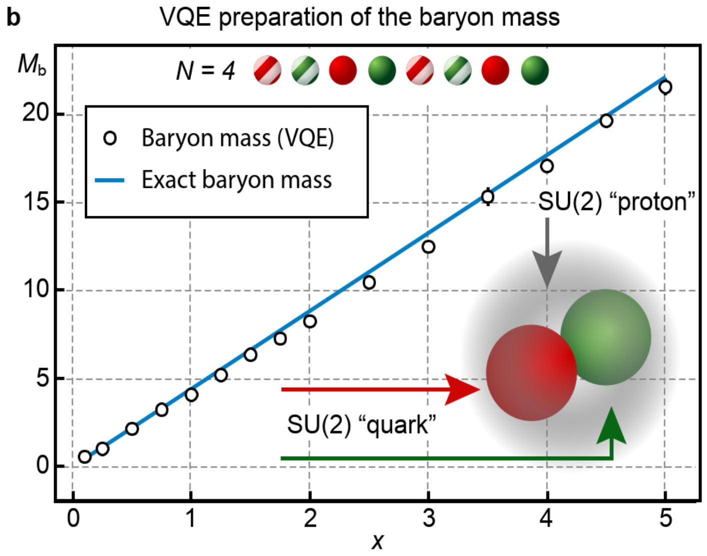
IBM-Q's superconducting qubits

Real-time evolution: Schwinger model,[1] SU(2),[2] SU(3),[3] ...

1+1D SU(3) gauge theory, one plaquette [3]

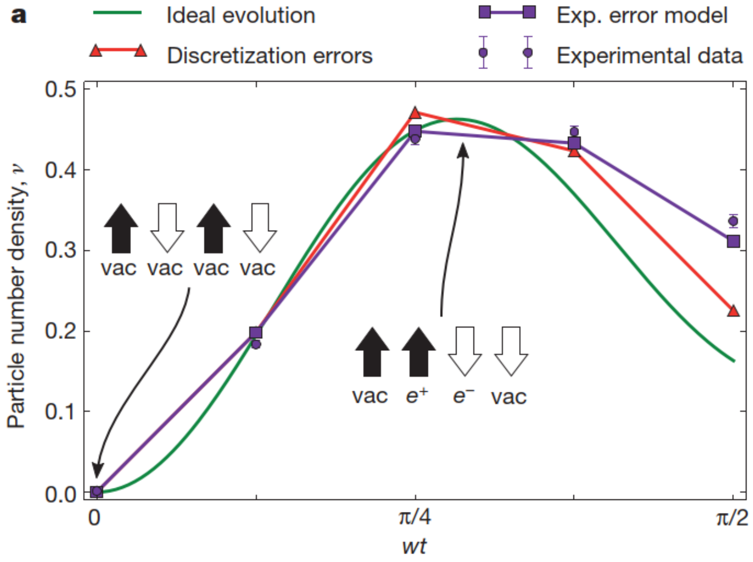


Variational computation: SU(2) "hadron" masses



Trapped ions

Real-time evolution: Schwinger model, [5]…



Martinez et al. (2016)

Cold atoms

Analog simulation: Bose-Hubbard,[6] Schwinger model,[7] ...

[1] Klco et al. (2018), de Jong et al. (2021), [2] Klco et al. (2019), [3] Ciavarella et al. (2019a,b), [4] Atas et al. (2021),
[5] Martinez et al. (2016), Nguyen et al. (2021), [6] Bloch et al. (2012), [7] Yang et al. (2020), Mil et al. (2020), ...

How to simulate these field theories?

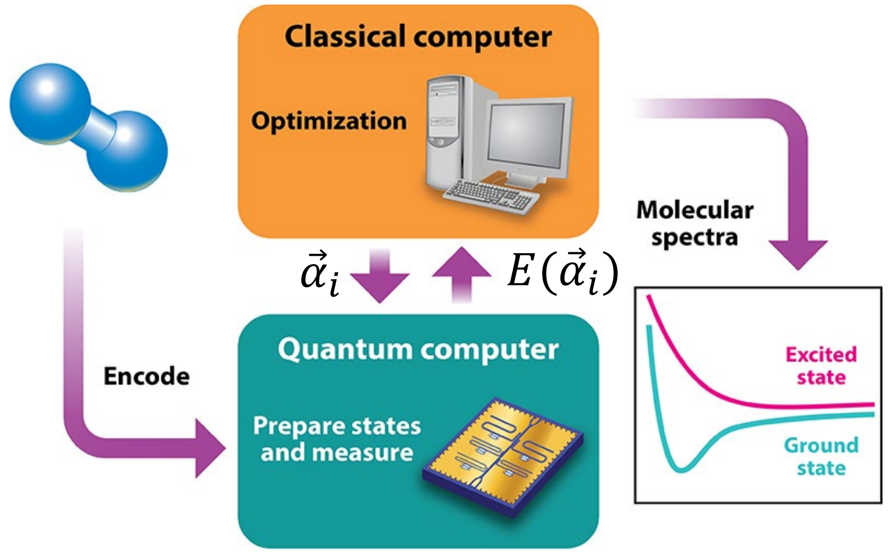
Example: hybrid quantum-classical algorithms

Key concept

Classical computer: main computation

Quantum computer: classically hard/intractable part

Advantages: already for small quantum hardware?



Sim et al. (2018)

[1] Peruzzo et al. (2014)

Variational Quantum Eigensolver (VQE) [1]

Goal

Find ground state and excited states of Hamiltonian $\mathcal{H}$

Variational approach

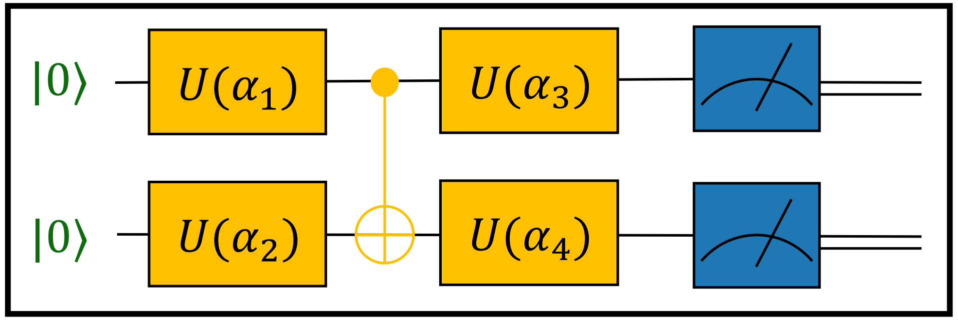
Minimize $E(\vec{\alpha}) = \langle\psi(\vec{\alpha})|\mathcal{H}|\psi(\vec{\alpha})\rangle$ w.r.t. parameters $\vec{\alpha}$

Classical computer

Given $E(\vec{\alpha}_i)$, find optimized parameters $\vec{\alpha}_{i+1}$

Quantum device

Given $\vec{\alpha}_i$, prepare $|\psi(\vec{\alpha})\rangle$ and measure $E(\vec{\alpha}_i)$

$|\psi(\vec{\alpha})\rangle =$ 

How to simulate these field theories?

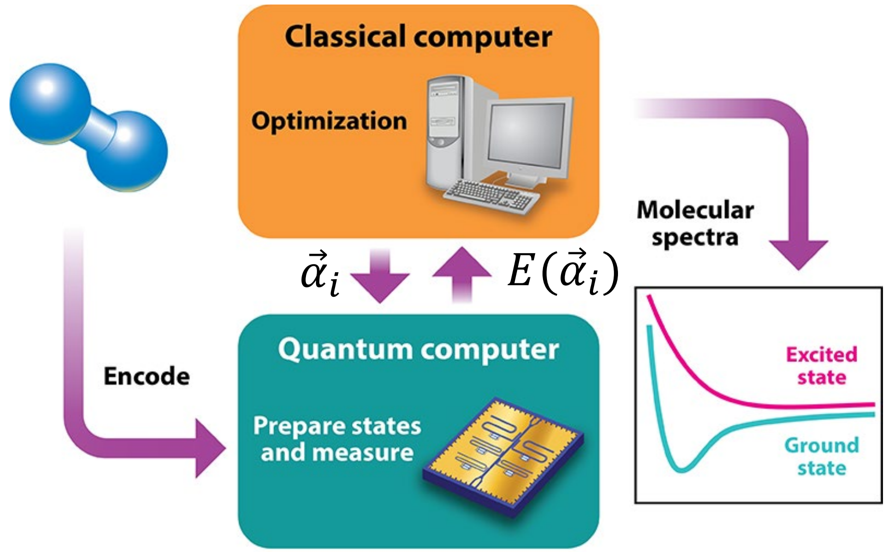
Example: hybrid quantum-classical algorithms	**Variational Quantum Eigensolver (VQE)** [1]

Key concept

Classical computer: main computation

Quantum computer: classically hard/intractable part

Advantages: already for small quantum hardware?



Sim et al. (2018)

Goal

Find ground state and excited states of Hamiltonian $\mathcal{H}$

Variational approach

Minimize $E(\vec{\alpha}) = \langle\psi(\vec{\alpha})|\mathcal{H}|\psi(\vec{\alpha})\rangle$ w.r.t. parameters $\vec{\alpha}$

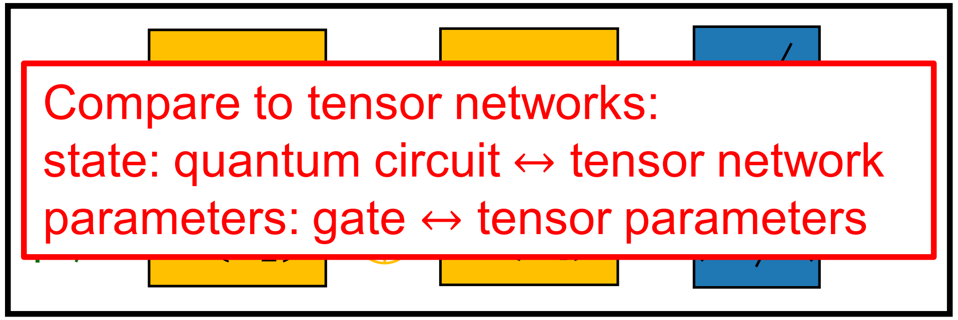
Classical computer

Given $E(\vec{\alpha}_i)$, find optimized parameters $\vec{\alpha}_{i+1}$

Quantum device

Given $\vec{\alpha}_i$, prepare $|\psi(\vec{\alpha})\rangle$ and measure $E(\vec{\alpha}_i)$

$|\psi(\vec{\alpha})\rangle =$



Compare to tensor networks:
state: quantum circuit ↔ tensor network
parameters: gate ↔ tensor parameters

[1] Peruzzo et al. (2014)

How to prepare the quantum state?

Quantum circuit design [1]

Maximal expressivity

$|\psi(\vec{\alpha})\rangle$ should reach all physical states in Hilbert space

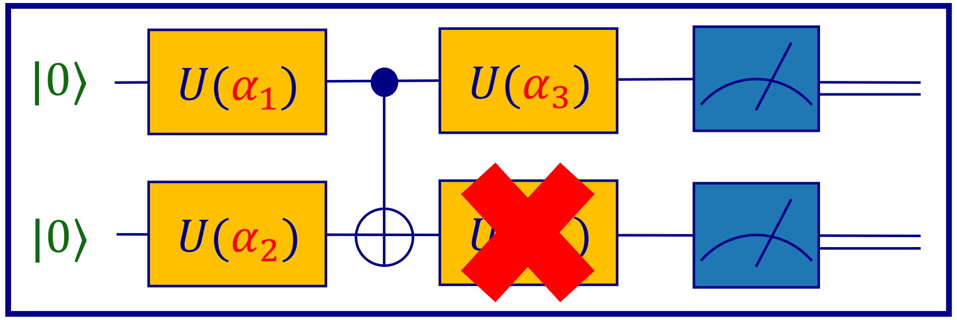
Minimality

$|\psi(\vec{\alpha})\rangle$ should not contain any redundant parameters

Symmetry

$|\psi(\vec{\alpha})\rangle$ should include physical symmetries

$|\psi(\vec{\alpha})\rangle =$



Example: geometrical method [2]

Manifolds

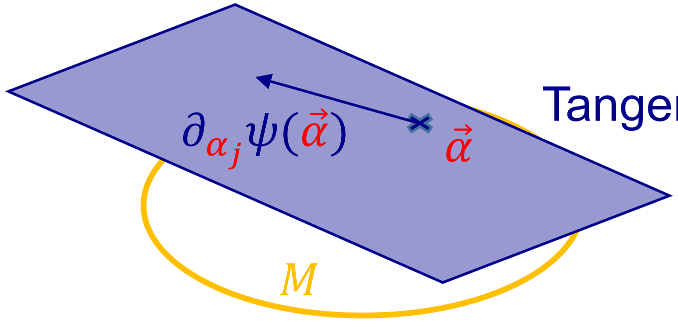
Circuit manifold M: states $|\psi(\vec{\alpha})\rangle$ reachable by circuit

State manifold S: states $|n(\vec{\alpha})\rangle$ of quantum device

Optimization

minimize: $\mathrm{codim}(M) = \dim(S) - \dim(M) \overset{!}{=} 0$

$$\det\begin{pmatrix} \mathrm{Re}|\partial_{\alpha_1}\psi\rangle & \cdots & \mathrm{Re}|\partial_{\alpha_k}\psi\rangle \\ \mathrm{Im}|\partial_{\alpha_1}\psi\rangle & \cdots & \mathrm{Im}|\partial_{\alpha_k}\psi\rangle \end{pmatrix} = 0 \text{ iff } \alpha_k \text{ redundant}$$



Tangent space of M at $\vec{\alpha}$

[1] Martinez et al. (2016), Klco et al. (2018,2019), Ciavarella et al. (2019a,b), Schweizer et al. (2019), Yang et al. (2020), Mil et al. (2020), de Jong et al. (2021), Nguyen et al. (2021), Atas et al. (2021), ..., [2] LF, Hartung, Jansen, Kühn, Stornati, *Quantum* (2021), *IEEE ICWS* (2021)

How to measure the energy on a quantum computer?

Example: massless Schwinger model

Original Hamiltonian

$$\mathcal{H} = -\frac{i}{2a}\sum_{n=0}^{N-2}\left(\psi_n^\dagger e^{i\theta_n}\psi_{n+1} - \mathrm{h.c.}\right) + \frac{ag^2}{2}\sum_{n=0}^{N-2} F_n^2$$

with $\theta_n = -aqA_n^1$, $gF_n = E_n$, $[\theta_n, L_m] = i\delta_{nm}$, $\theta_n \in [0,2\pi]$

Eliminate θ_n

$\psi_n^\dagger e^{i\theta_n}\psi_{n+1} \to \psi_n^\dagger \psi_{n+1}$ from transformation:

$\psi_n \to (\prod_{k=0}^{n-1} e^{-i\theta_n})\psi_n$ and $\psi_n^\dagger \to \psi_n^\dagger(\prod_{k=0}^{n-1} e^{i\theta_{n-k}})$

Eliminate F_n

$F_n = \sum_{k=0}^{n} Q_k$ from solving Gauß law (for OBC):

$F_n - F_{n-1} = Q_n \; \forall n$, where $Q_n = \psi_n^\dagger\psi_n - \frac{1}{2}[1-(-1)^n]$

[1] Banks et al. (1976), Hamer et al. (1997)

Mapping the model to qubits

Dimensionless spin Hamiltonian [1]

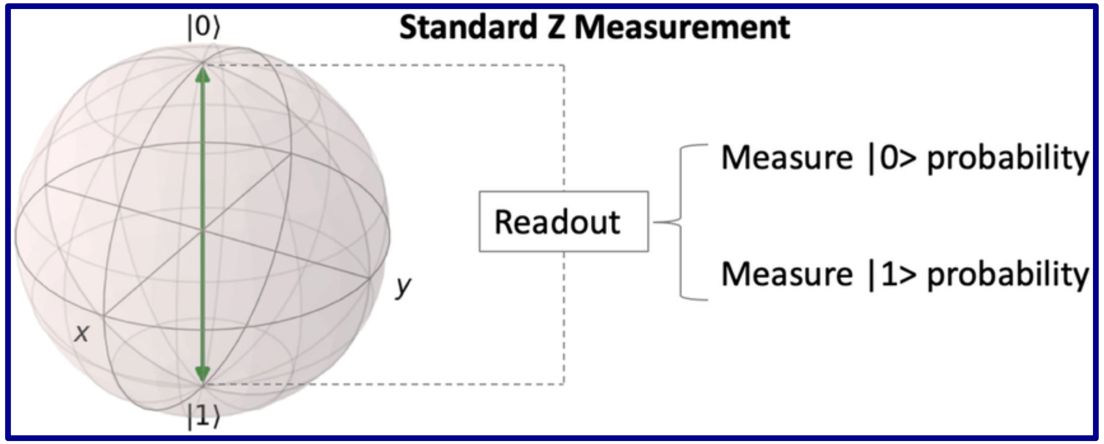
$$\mathcal{H} = x\sum_{n=0}^{N-2}(\sigma_n^+\sigma_{n+1}^- + \sigma_n^-\sigma_{n+1}^+) + \frac{1}{2}\sum_{n=0}^{N-2}\left\{\sum_{k=0}^{n}[(-1)^k + \sigma_k^z]\right\}^2$$

from mapping $\psi_n^\dagger\psi_{n+1} \to \sigma_n^+\sigma_{n+1}^-$ and $\phi_n^\dagger\phi_n \to \frac{1}{2}(\sigma_n^z + \mathbb{I})$

Quantum computer

Measurement of $\langle\psi|\boldsymbol{O}|\psi\rangle$ with $\boldsymbol{O} \in \{\mathbb{I}, \sigma^z\}^{\otimes N}$

$\mathcal{H} = \sum_k h_k U_k^* \boldsymbol{O}_{\boldsymbol{k}} U_k$ with $U_k^*\boldsymbol{O}_{\boldsymbol{k}}U_k \in \{\mathbb{I}, \sigma^x, \sigma^y, \sigma^z\}^{\otimes N}$



Gokhale et al. (2020)

How to deal with gauge fields?

Infinite Hilbert space

Problem

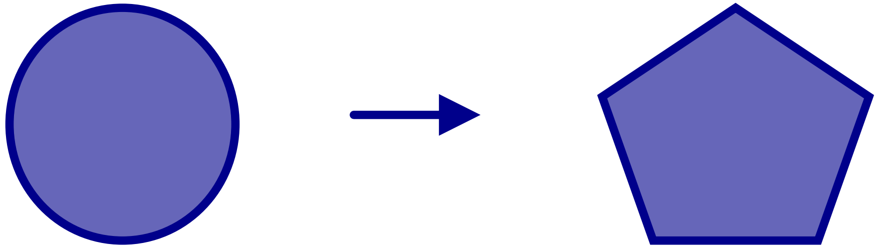
Continuous gauge theory requires ∞-dim. Hilbert space

First approach

Integrate out gauge field: only possible in 1+1D

Second approach

Approximate gauge group:[1] e.g. $U(1) \rightarrow \mathbb{Z}_n$



Third approach

Truncate irreps:[2] e.g. for $F_j|l\rangle = |l\rangle$, use finite $|l| < L$

Many more approaches…

Gauge invariance

Problem

Gauge invariance requires imposing local constraints

First approach

Penalize unphysical states,[3] e.g. $\mathcal{H}_{\text{penalty}} = \lambda\left(\sum_{j=1}^{N} Q_j\right)^2$

Second approach

Analytically solve Gauß law at every site [4]

Third approach

Gauge-invariant formulation, e.g. loop-string-hadron [5]

Many more approaches…

[1] Zohar et al. (2013), ..., [2] Horn (1981), ..., [3] Banerjee et al. (2012), ..., [4] Klco et al. (2018), ..., [5] Raychowdhury, Stryker (2020)

Outlook: how can we address the sign problem in 3+1D?

Example: U(1) lattice gauge theory with θ-term

Goal

Study phase transition at $\theta = \pi$ and large $g = \beta^{-1/2}$

Theoretical requirements

Derive 3+1D θ-term in Hamiltonian formulation [1]

Develop resource-efficient quantum algorithms for lattice gauge theories in 1+1D,[2] 2+1D,[3] and finally 3+1D

First classical computations

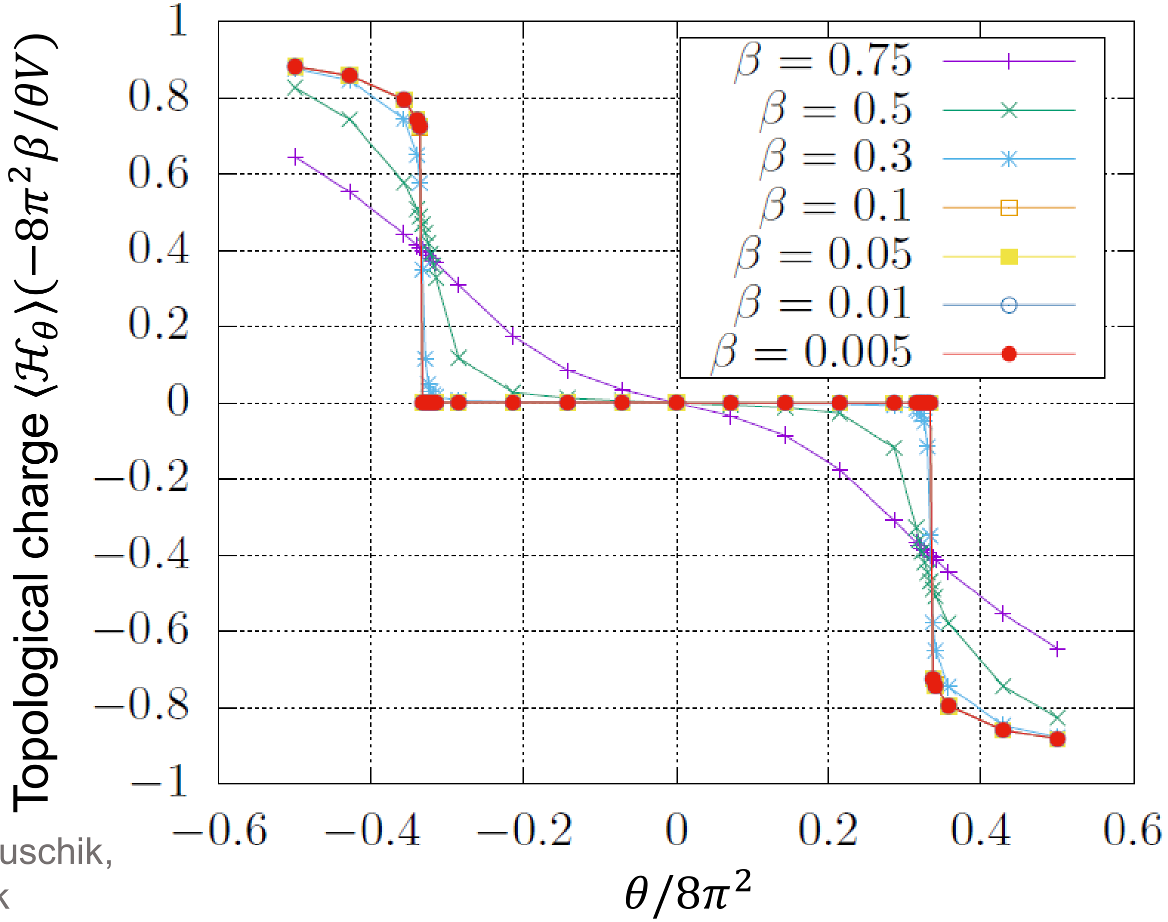
Study phase transition with exact diagonalization [1] →

Future work

Larger volumes: tensor-network and *quantum* simulations

[1] Kan, LF, Kühn, Zhan, Haase, Muschik, Jansen, PRD (2021)
[2] Ferguson, ..., Jansen, PRL (2021), ..., and ongoing work
[3] Haase, ..., Jansen, Muschik, Quantum (2021), Paulson, ..., Jansen, Zoller, Muschik, PRX Quantum (2021), Clemente, Crippa, Jansen (2022), ..., and ongoing work

First classical results for a single cube



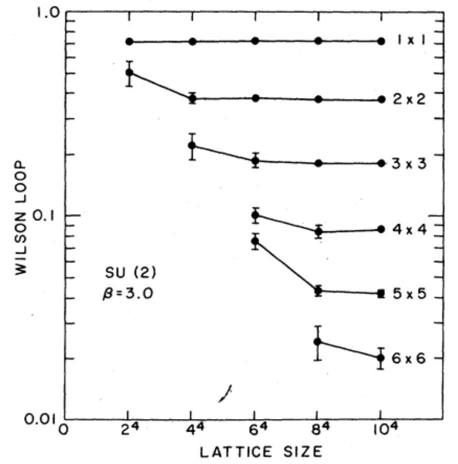
Summary: where do we stand, where will we go?

The Path to Go…

Present

Hardware: $\mathcal{O}(10-100)$ qubits with error mitigation

Algorithms: first QC implementations of 1+1D LGTs, first resource-efficient formulations of 2+1D LGTs



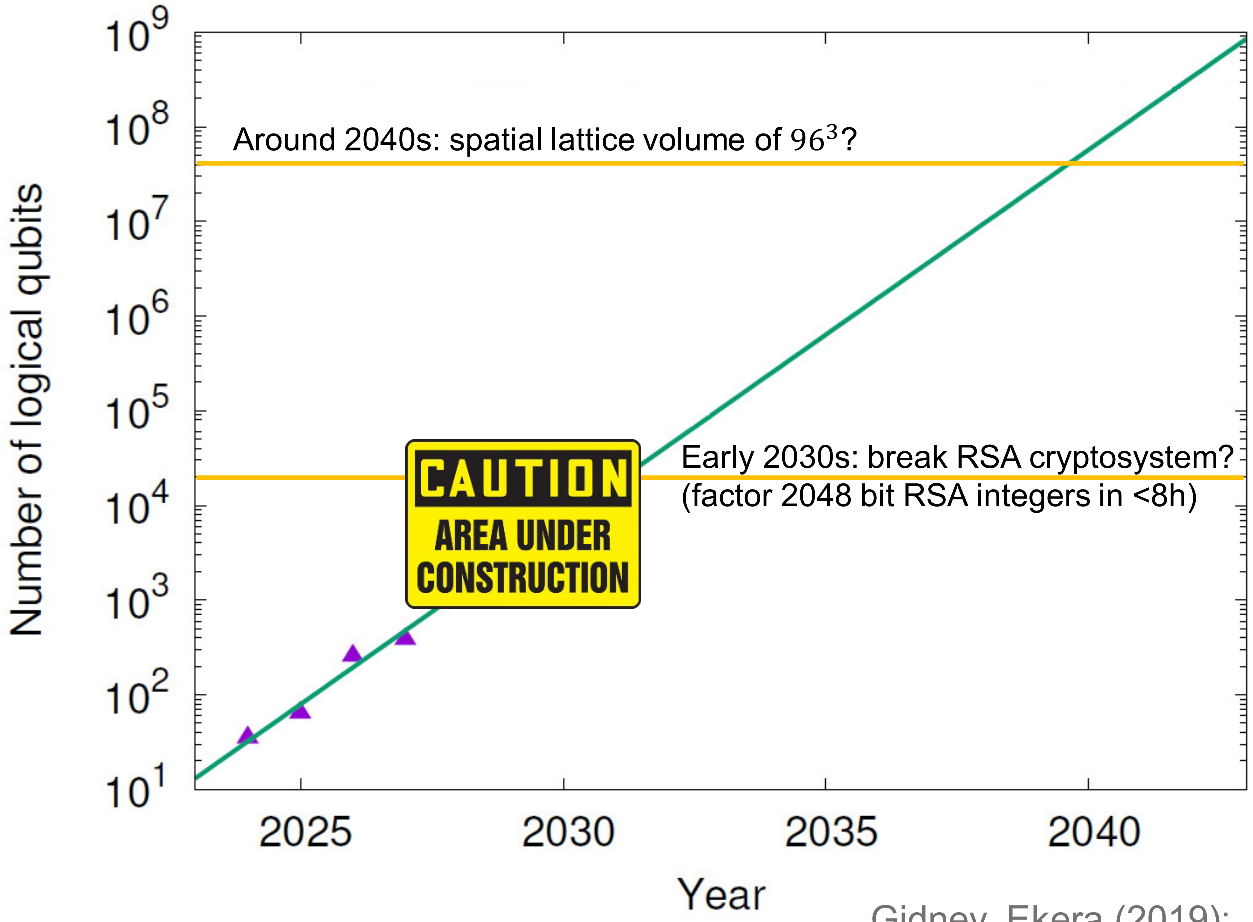
Creutz (1980)

Future

Hardware: $\mathcal{O}(1000)$ error-corrected qubits by 2029?

Algorithms: improved Hamiltonians, resource-efficient formulations of 3+1D LGTs, …

A Rough Sketch…



Gidney, Ekera (2019); Kan, Nam (2021)

Summary: where do we stand, where will we go?

The Way Forward…

Quantum

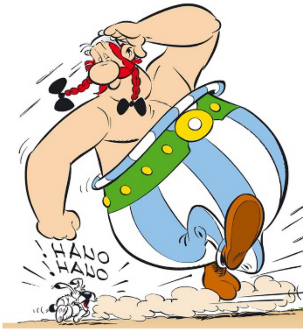
new and quickly progressing

→ high risk, high gain

Classical

more established but limited

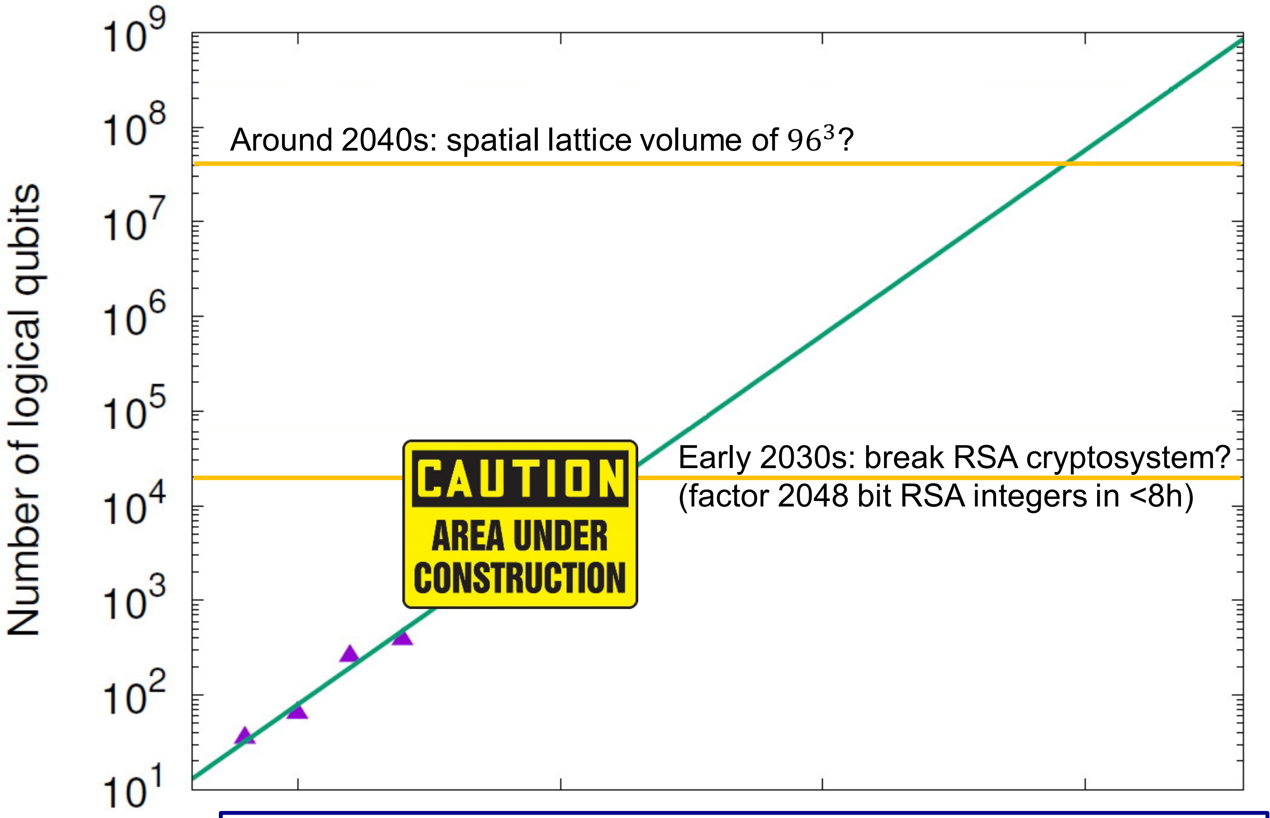
→ solid, promising



Combination

→ quantum-classical algorithms

A Rough Sketch…



Thanks for listening!

Do you have any questions?