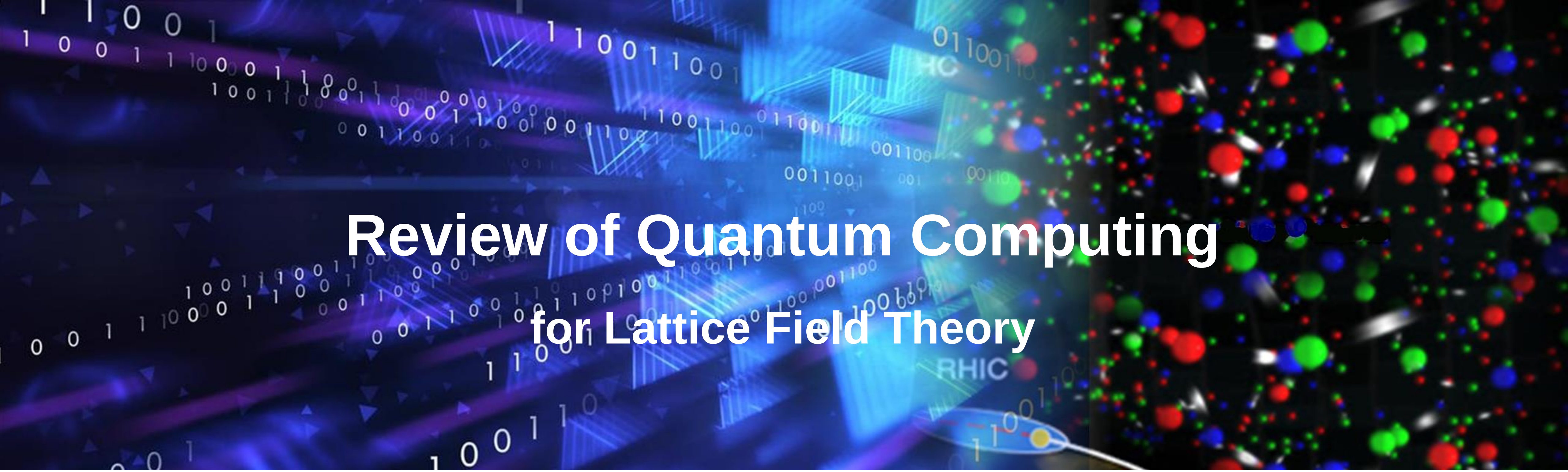


Review of Quantum Computing

for Lattice Field Theory

Lena Funcke

with Karl Jansen, Stefan Kühn, Tobias Hartung, et al.

Lattice 2022 Conference, Bonn, 8 August 2022

Why quantum computing?

Computational costs of lattice field theory	Computational challenges of lattice field theory

Why quantum computing?

Computational costs of lattice field theory | **Computational challenges of lattice field theory**

Supercomputer usage for different fields (INCITE 2019)

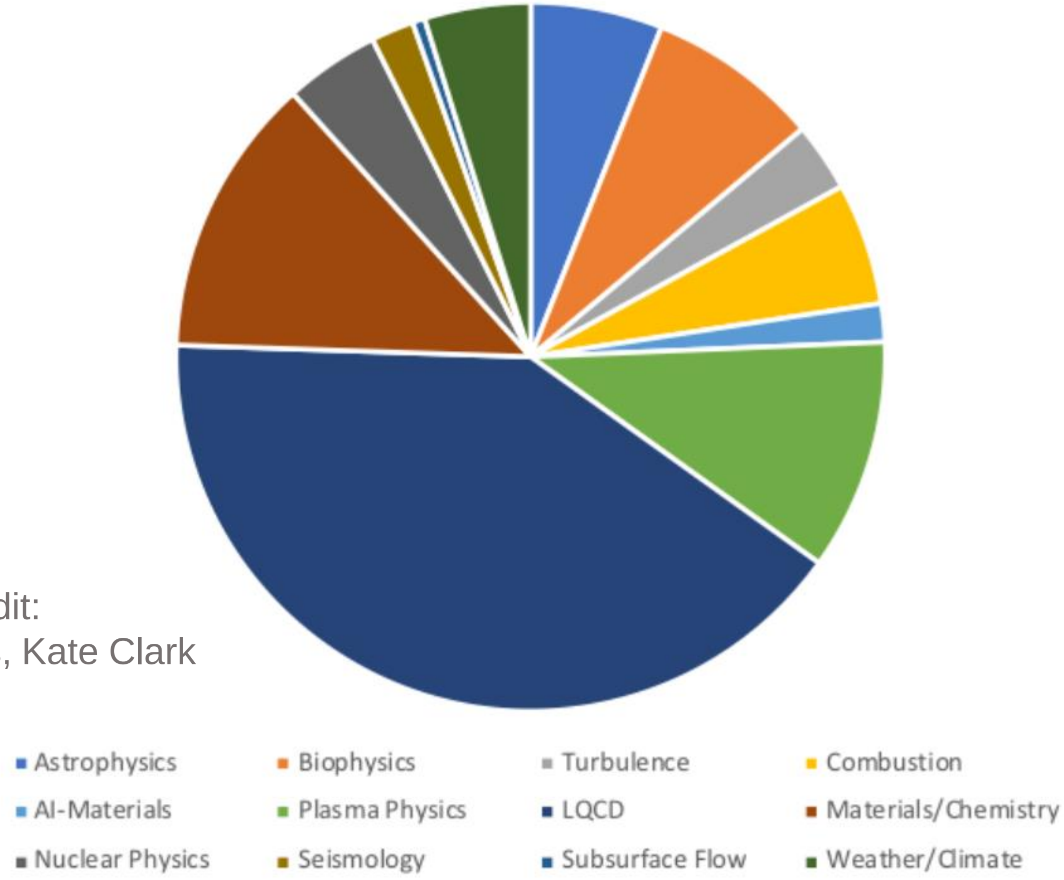


Figure credit:
Jack Wells, Kate Clark

Why quantum computing?

Computational costs of lattice field theory | **Computational challenges of lattice field theory**

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD: $\sim$ 40%**

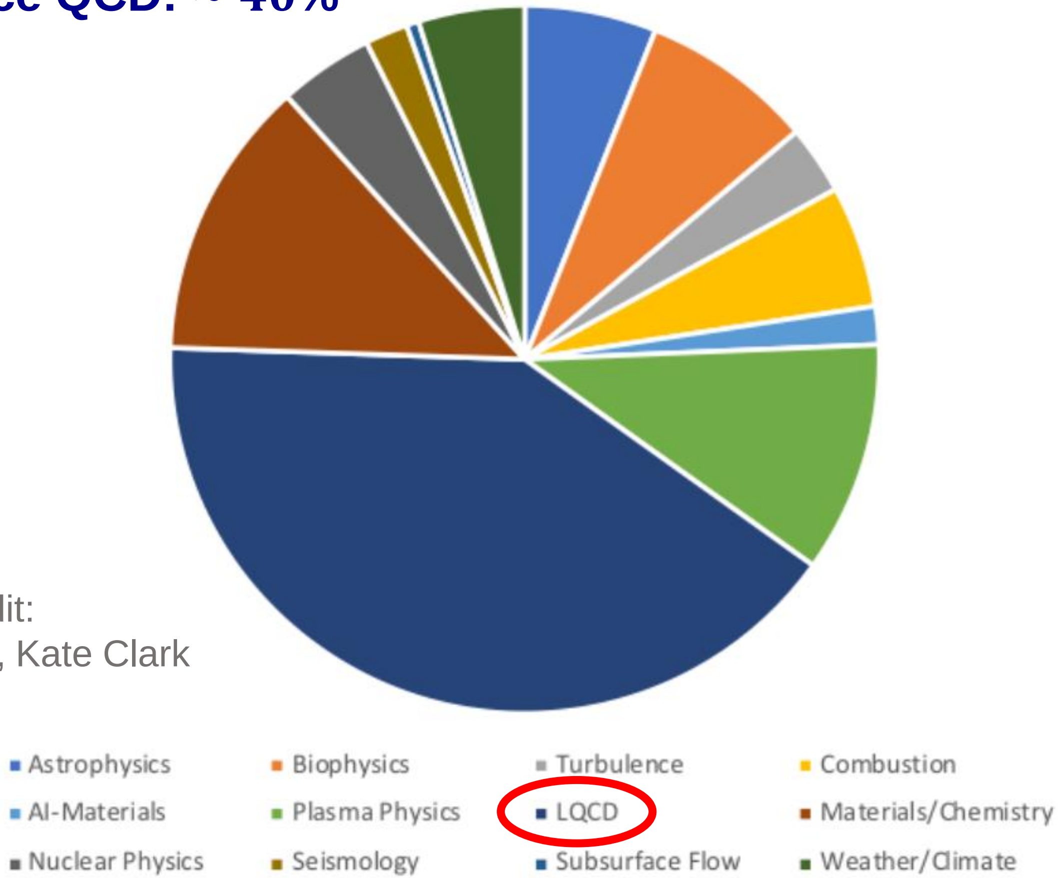


Figure credit:
Jack Wells, Kate Clark

Why quantum computing?

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD: ~ $\mathbf{40\%}$**

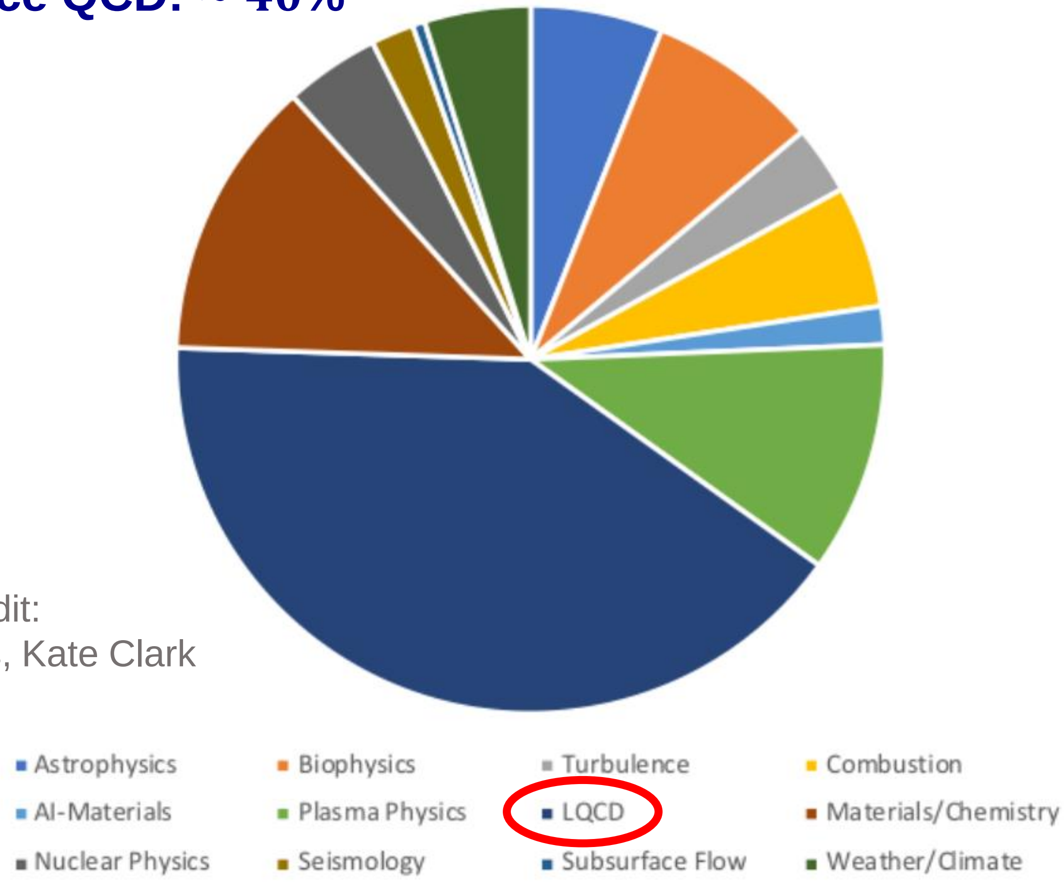


Figure credit:
Jack Wells, Kate Clark

Computational challenges of lattice field theory

Critical slowing down, large autocorrelation times, …

→ Machine learning (*Algorithms* 8/10 Aug.)

Why quantum computing?

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD: ~ 40%**

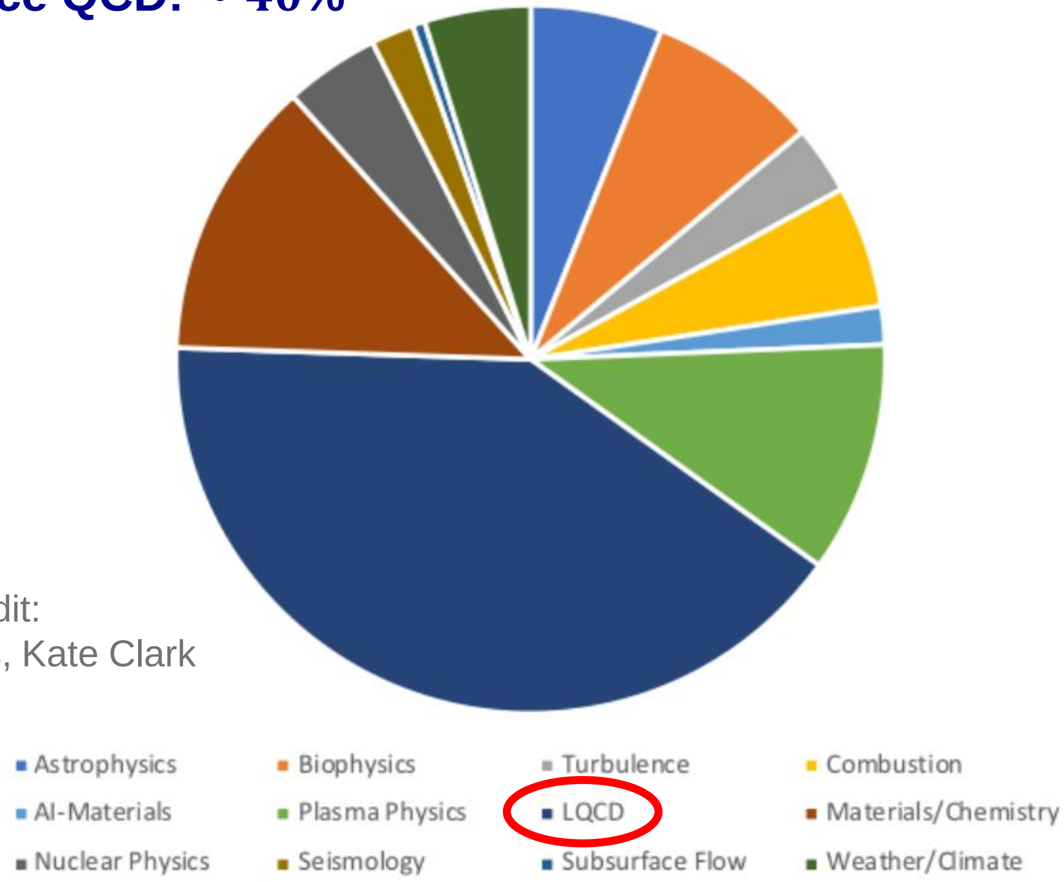


Figure credit:
Jack Wells, Kate Clark

Computational challenges of lattice field theory

Critical slowing down, large autocorrelation times, …

→ Machine learning (*Algorithms* 8/10 Aug.)

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

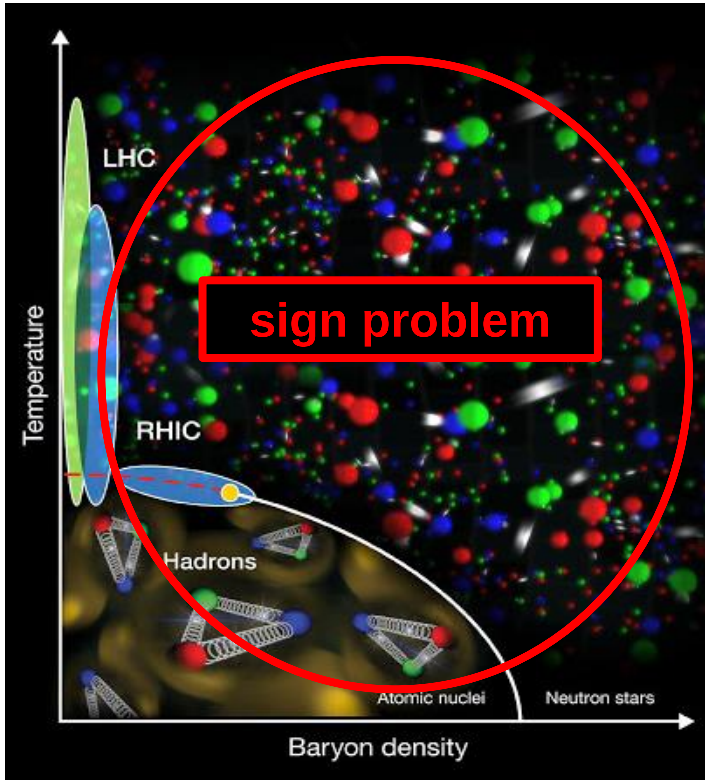


Figure credit:
BNL/RHIC

Why quantum computing?

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD: ~ 40%**

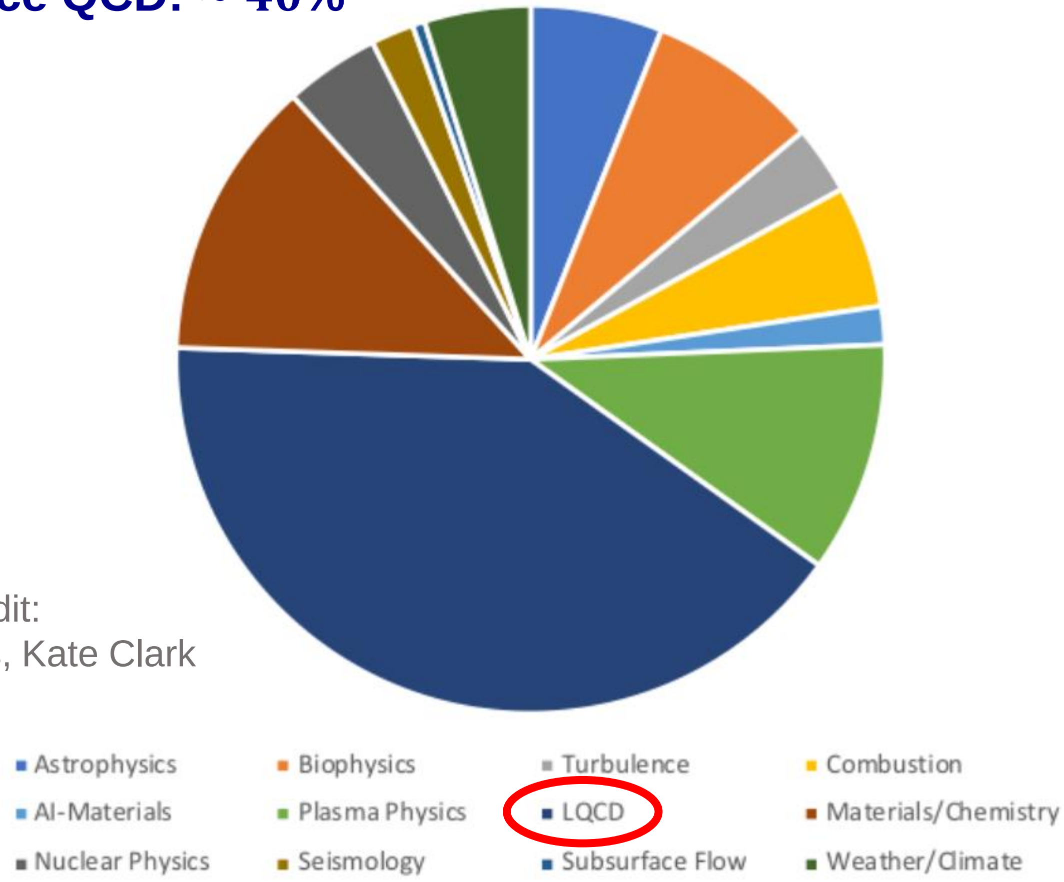


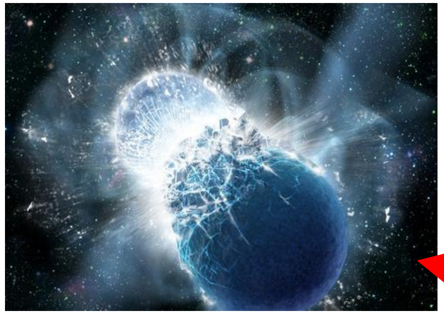
Figure credit:
Jack Wells, Kate Clark

Computational challenges of lattice field theory

Critical slowing down, large autocorrelation times, …

→ Machine learning (*Algorithms* 8/10 Aug.)

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



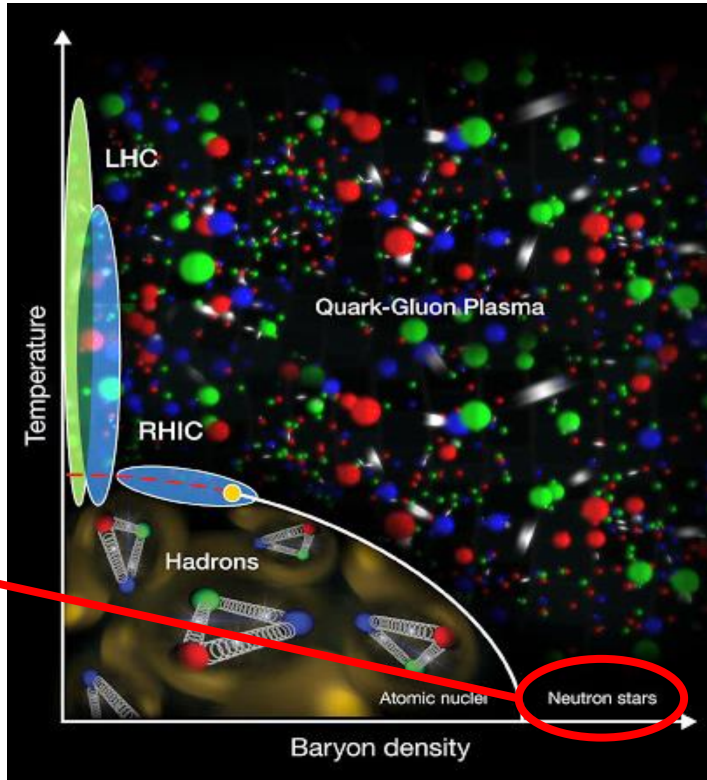


Figure credit:
BNL/RHIC, CfA

Why quantum computing?

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD: ~ 40%**

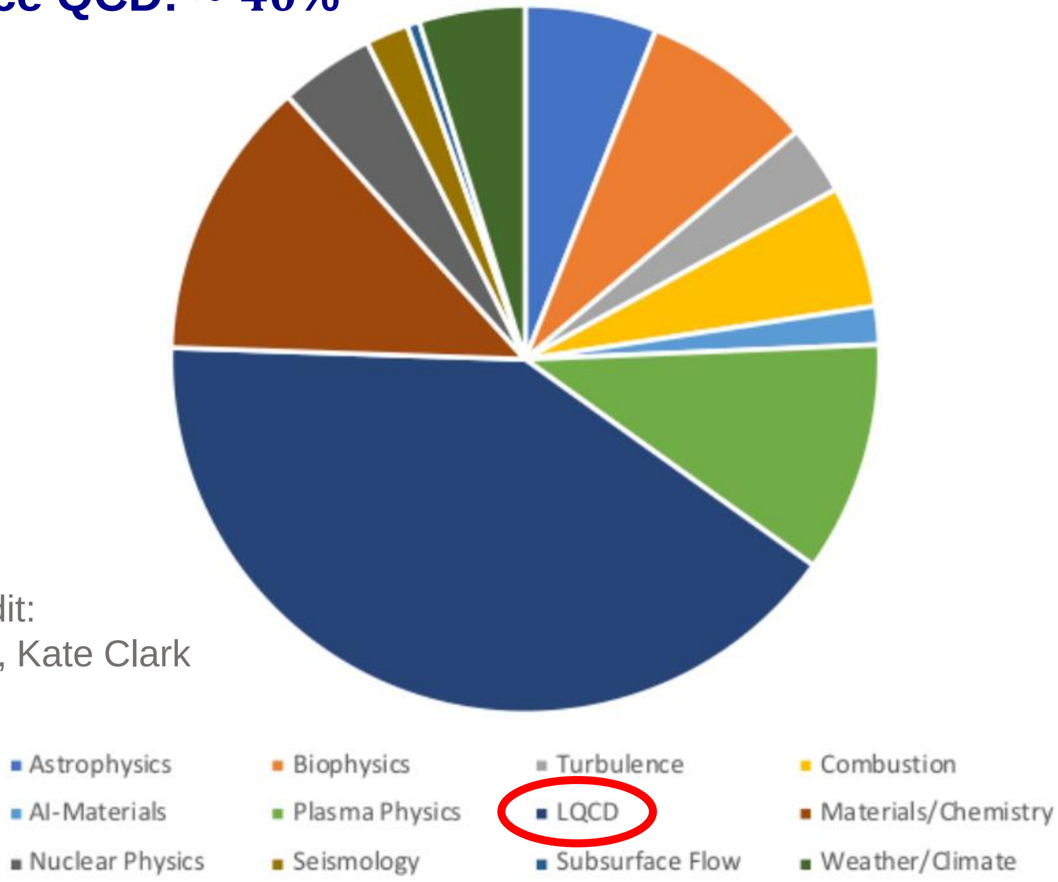


Figure credit:
Jack Wells, Kate Clark

Computational challenges of lattice field theory

Critical slowing down, large autocorrelation times, …

→ Machine learning (*Algorithms* 8/10 Aug.)

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

→ Tensor networks
(*Plenary* 13 Aug. and
Algorithms 9/10 Aug.)

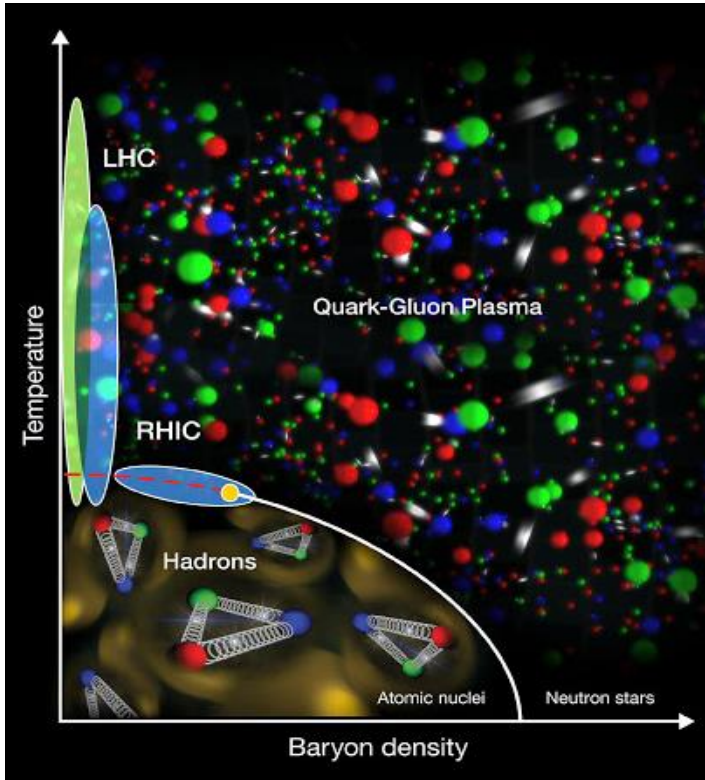


Figure credit:
BNL/RHIC

Why quantum computing?

Computational costs of lattice field theory

Supercomputer usage for different fields (INCITE 2019)

→ **Lattice QCD: ~ 40%**

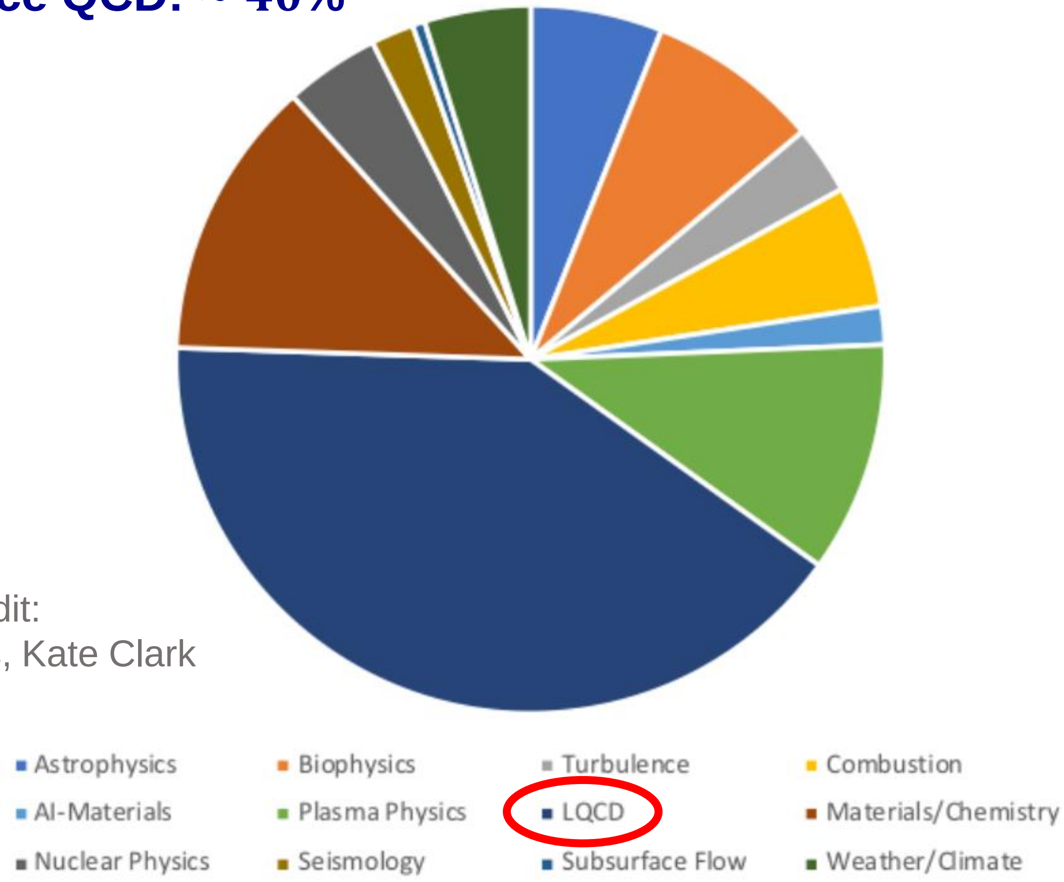


Figure credit:
Jack Wells, Kate Clark

Computational challenges of lattice field theory

Critical slowing down, large autocorrelation times, …

→ Machine learning (*Algorithms* 8/10 Aug.)

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

→ Tensor networks (*Plenary* 13 Aug. and *Algorithms* 9/10 Aug.)

→ **Quantum computing** (*Algorithms* 9/10/11 Aug.)

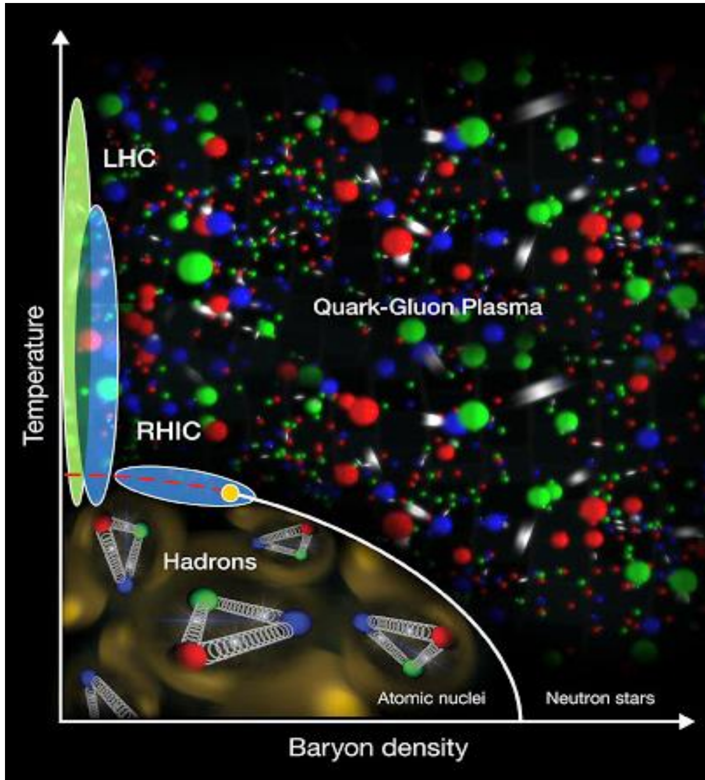


Figure credit:
BNL/RHIC

Why is the sign problem exponentially hard?

Example: finite baryon chemical potential	Reweighting procedure

Why is the sign problem exponentially hard?

Example: finite baryon chemical potential | **Reweighting procedure**

Partition function

$Z = \int DUD\bar{\psi}\, D\psi e^{-S} = \int DUe^{-S_g} \det M$

Why is the sign problem exponentially hard?

Example: finite baryon chemical potential | **Reweighting procedure**

Partition function

$Z = \int DUD\bar{\psi}\, D\psi e^{-S} = \int DUe^{-S_g} \det M$

Importance sampling

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

Why is the sign problem exponentially hard?

Example: finite baryon chemical potential	Reweighting procedure

Partition function

$Z = \int DUD\bar{\psi}\, D\psi e^{-S} = \int DUe^{-S_g} \det M$

Importance sampling

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

Sign problem

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

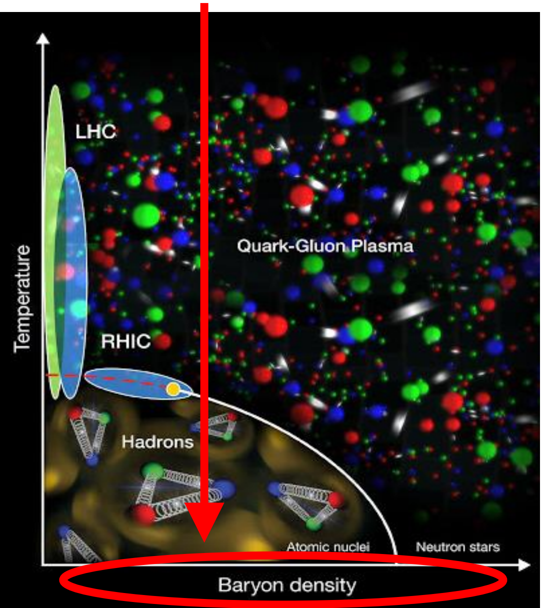


Figure credit:
BNL/RHIC

Why is the sign problem exponentially hard?

Example: finite baryon chemical potential

Partition function

$Z = \int DUD\bar{\psi}\, D\psi e^{-S} = \int DU e^{-S_g} \det M$

Importance sampling

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

Sign problem

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

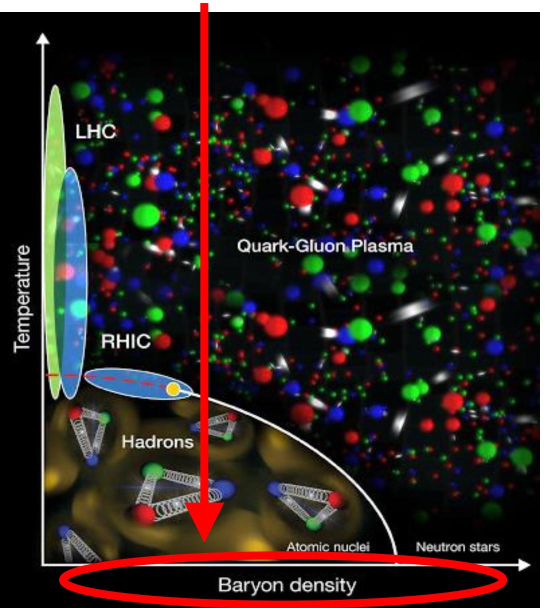


Figure credit:
BNL/RHIC

Reweighting procedure

Example: phase quenched theory

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

Why is the sign problem exponentially hard?

Example: finite baryon chemical potential

Partition function

$Z = \int DUD\bar{\psi}\, D\psi e^{-S} = \int DUe^{-S_g} \det M$

Importance sampling

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

Sign problem

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

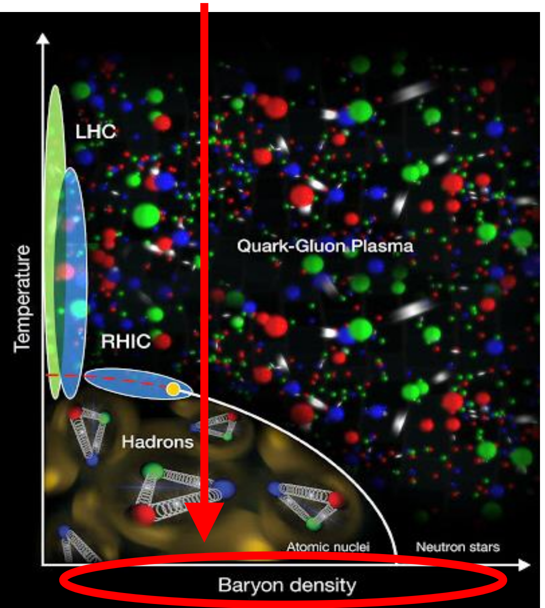


Figure credit: BNL/RHIC, de Forcrand

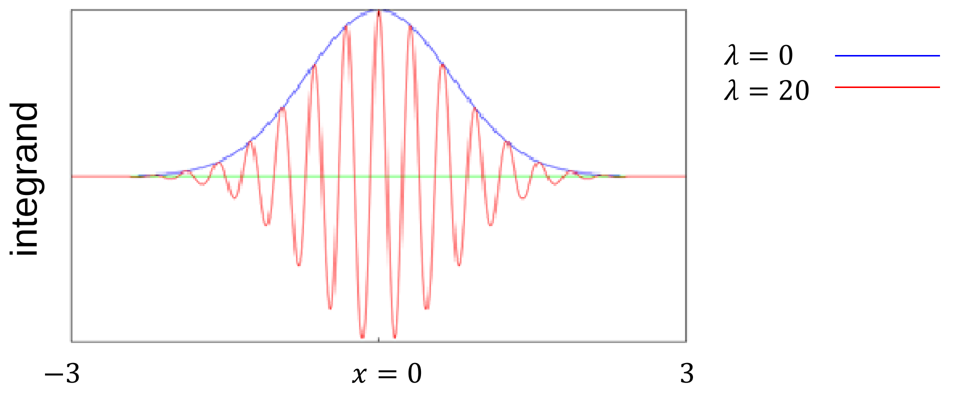
Reweighting procedure

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



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

Why is the sign problem exponentially hard?

Example: finite baryon chemical potential

Partition function

$Z = \int DUD\bar{\psi}\, D\psi e^{-S} = \int DUe^{-S_g} \det M$

Importance sampling

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

Sign problem

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

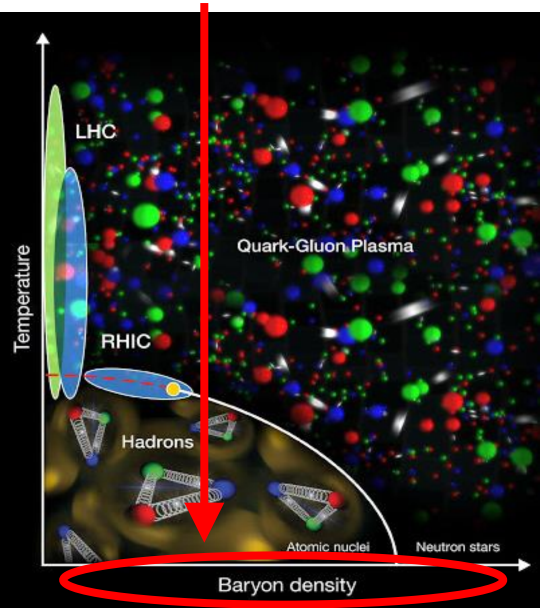


Figure credit:
BNL/RHIC,
de Forcrand

Reweighting procedure

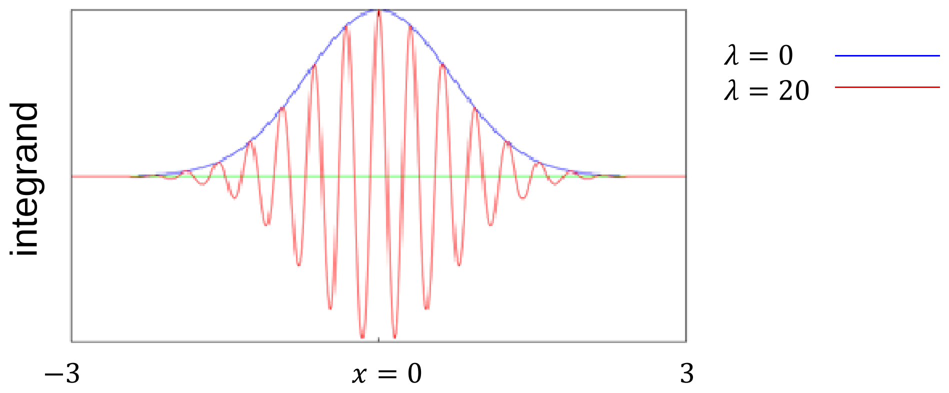
Example: phase quenched theory

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

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	Why is(n’t) classical computing enough?

Do we really need *quantum* computing?

Classical approaches to tackle the sign problem	**Why is(n't) classical computing enough?**

Tensor networks

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

Do we really need *quantum* computing?

Classical approaches to tackle the sign problem | **Why is(n’t) classical computing enough?**

Tensor networks

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

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

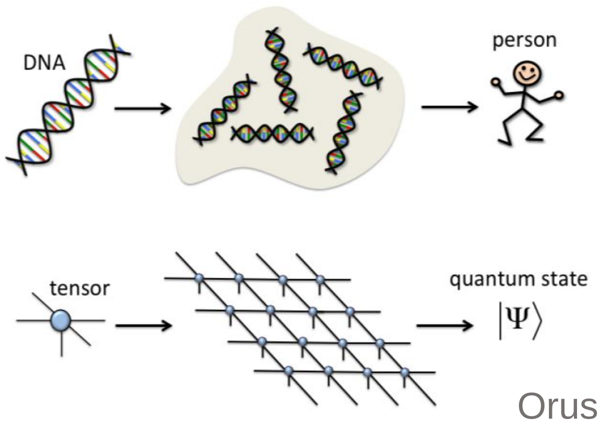
Do we really need *quantum* computing?

Classical approaches to tackle the sign problem | **Why is(n't) classical computing enough?**

Tensor networks

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

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



Orus (2014)

Do we really need *quantum* computing?

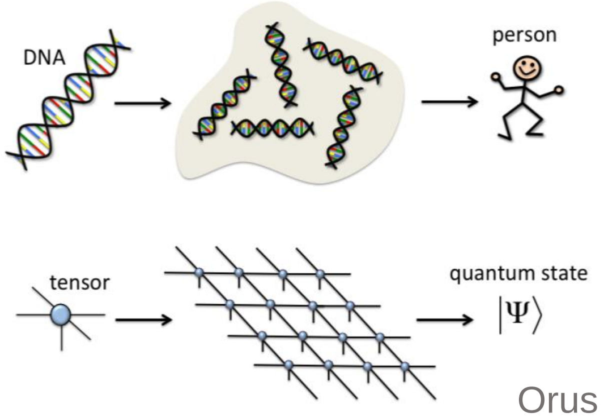
Classical approaches to tackle the sign problem | **Why is(n't) classical computing enough?**

Tensor networks

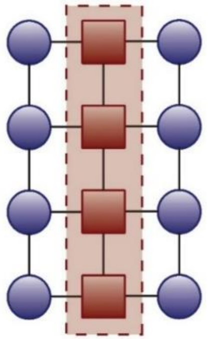
Describe quantum state $|\psi\rangle$ by network of small tensors

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

E.g. variational algorithm: minimize energy $E = \langle\psi|\mathcal{H}|\psi\rangle$



Orus (2014)



Bañuls et al. (2013)

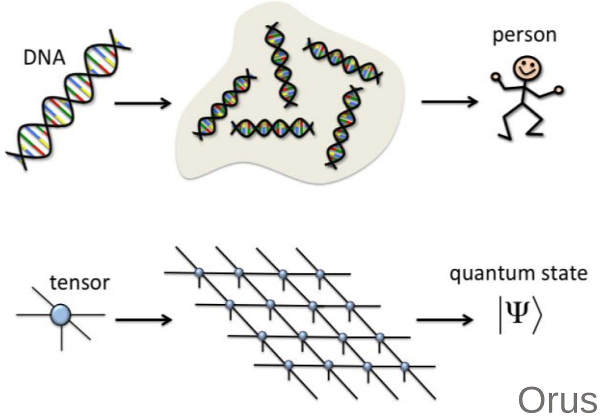
Do we really need *quantum* computing?

Tensor networks

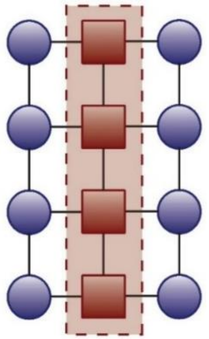
Describe quantum state $|\psi\rangle$ by network of small tensors

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

E.g. variational algorithm: minimize energy $E = \langle\psi|\mathcal{H}|\psi\rangle$



Orus (2014)



Bañuls et al. (2013)

Other approaches

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

[1] Wynen et al. (2020), ...

Do we really need *quantum* computing?

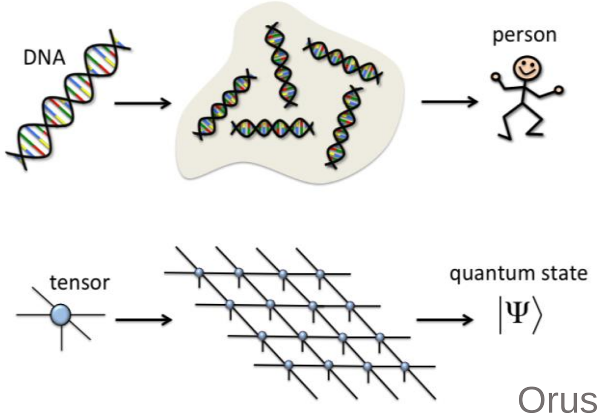
Classical approaches to tackle the sign problem	Why is(n't) classical computing enough?

Tensor networks

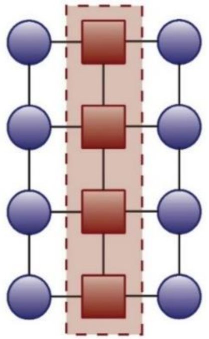
Describe quantum state $|\psi\rangle$ by network of small tensors

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

E.g. variational algorithm: minimize energy $E = \langle\psi|\mathcal{H}|\psi\rangle$



Orus (2014)

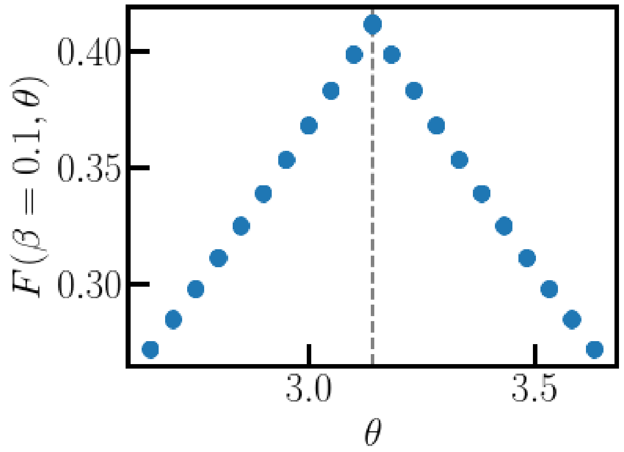

Bañuls et al. (2013)

Other approaches

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

Prospects

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



Nakayama et al. (2022)

[1] Wynen et al. (2020), ..., [2] Bañuls et al. (2017), Byrnes et al. (2002), Pichler et al. (2016), ...

Do we really need *quantum* computing?

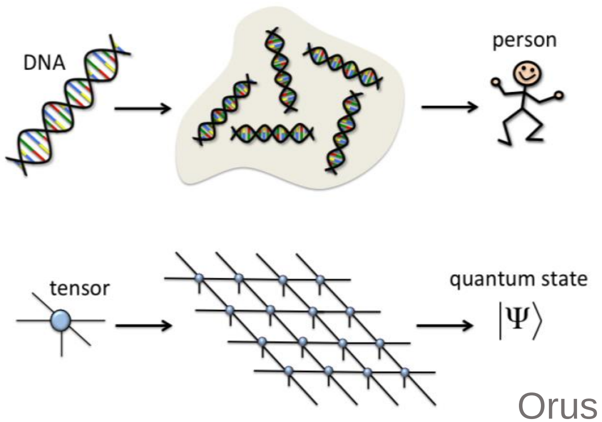
Classical approaches to tackle the sign problem

Tensor networks

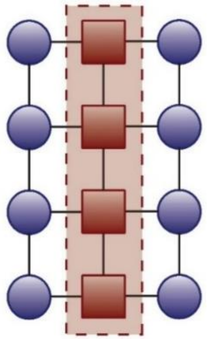
Describe quantum state $|\psi\rangle$ by network of small tensors

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

E.g. variational algorithm: minimize energy $E = \langle\psi|\mathcal{H}|\psi\rangle$



Orus (2014)



Bañuls et al. (2013)

Other approaches

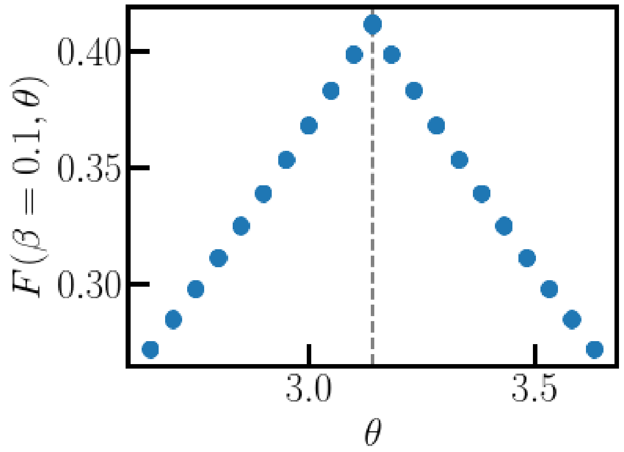
Deep learning for path integral contour deformations[1] ...

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]



Nakayama et al. (2022)

[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?

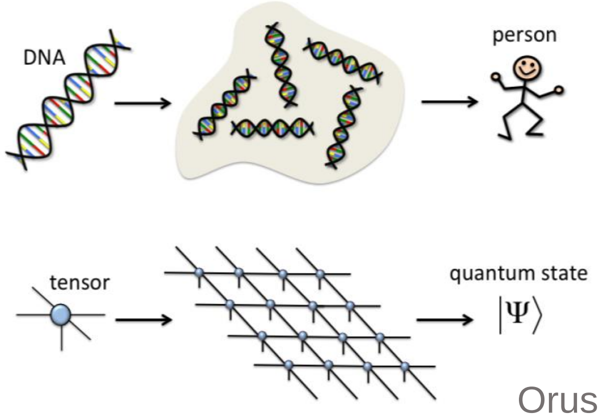
Classical approaches to tackle the sign problem

Tensor networks

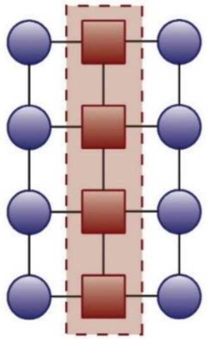
Describe quantum state $|\psi\rangle$ by network of small tensors

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

E.g. variational algorithm: minimize energy $E = \langle\psi|\mathcal{H}|\psi\rangle$



Orus (2014)



Bañuls et al. (2013)

Other approaches

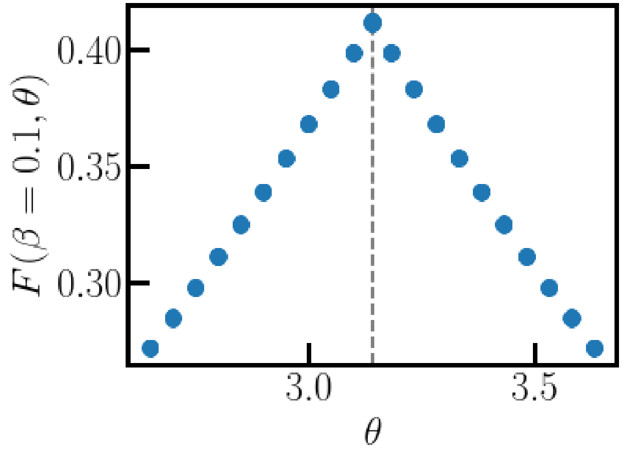
Deep learning for path integral contour deformations[1] ...

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]



Nakayama et al. (2022)

Challenges

No efficient parametrization of highly entangled states

[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?

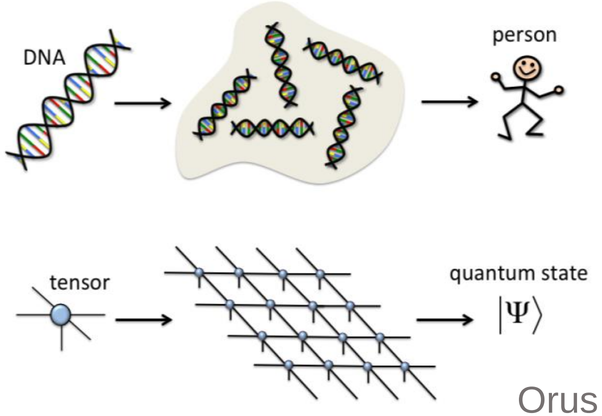
Classical approaches to tackle the sign problem

Tensor networks

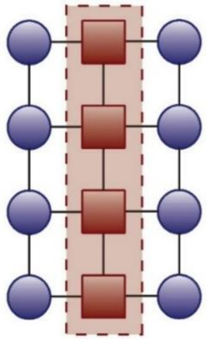
Describe quantum state $|\psi\rangle$ by network of small tensors

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

E.g. variational algorithm: minimize energy $E = \langle\psi|\mathcal{H}|\psi\rangle$



Orus (2014)



Bañuls et al. (2013)

Other approaches

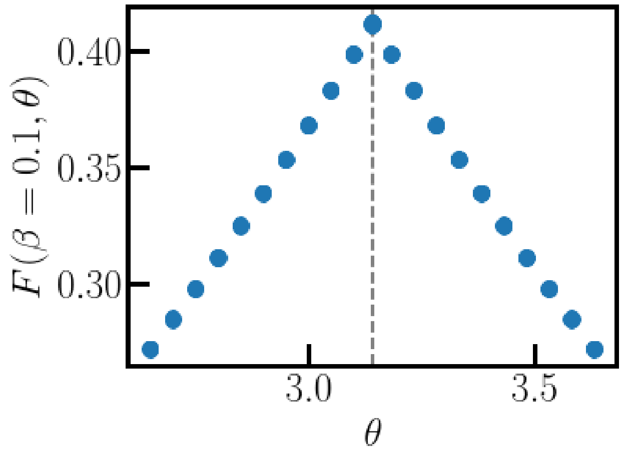
Deep learning for path integral contour deformations[1] ...

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]



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?

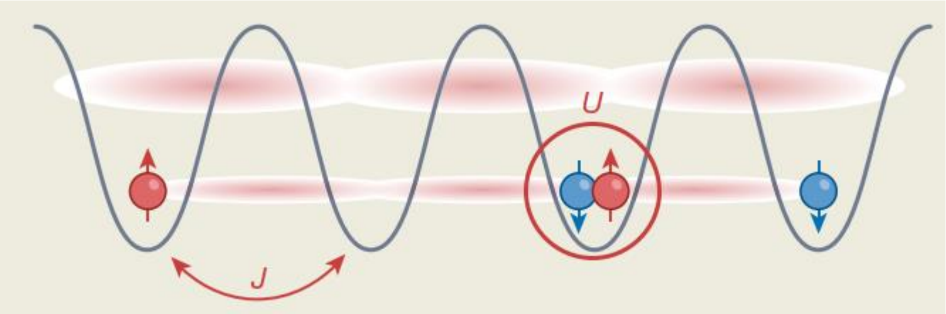
Example: 1+1D Bose-Hubbard model	**Why is(n't) classical computing enough?**

Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model	Why is(n't) classical computing enough?

Hamiltonian

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

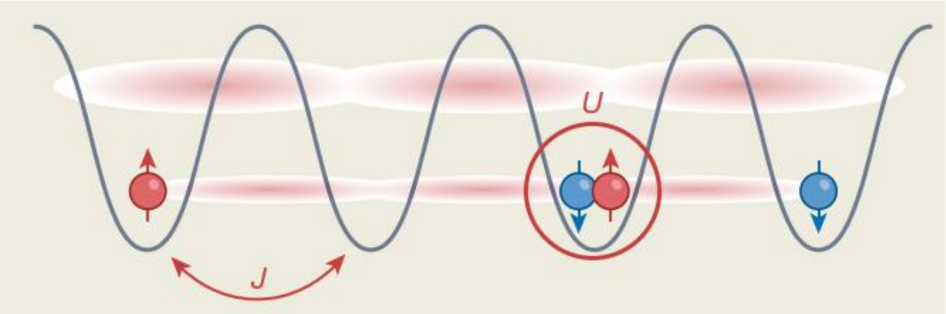


Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model	Why is(n't) classical computing enough?

Hamiltonian

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



Real-time simulation [1]

Analog quantum simulator: ultracold atoms

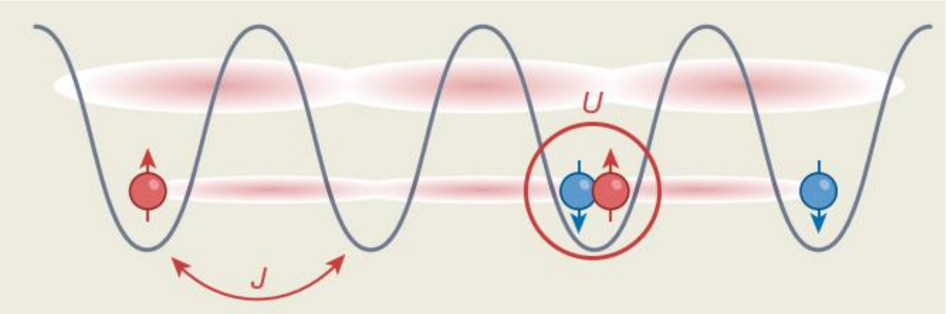
[1] Trotzky et al. (2012)

Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model | **Why is(n't) classical computing enough?**

Hamiltonian

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



Real-time simulation [1]

Analog quantum simulator: ultracold atoms

Classical benchmark: tensor networks (MPS)

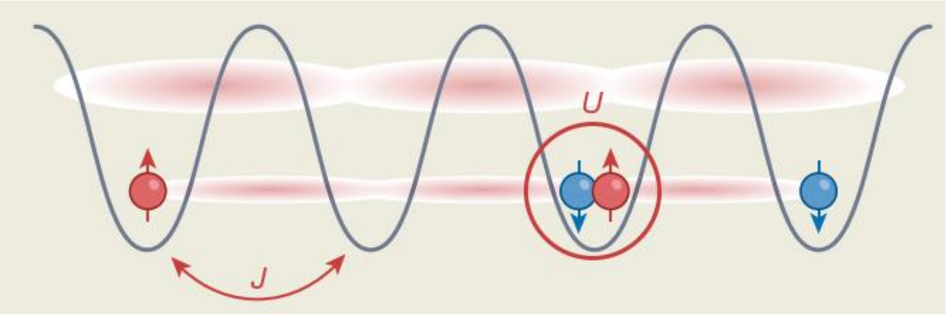
[1] Trotzky et al. (2012)

Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model	Why is(n't) classical computing enough?

Hamiltonian

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



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]

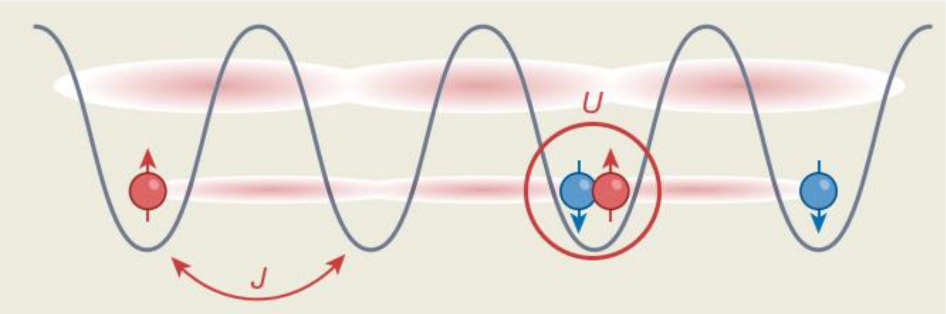
[1] Trotzky et al. (2012)

Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model

Hamiltonian

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



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)

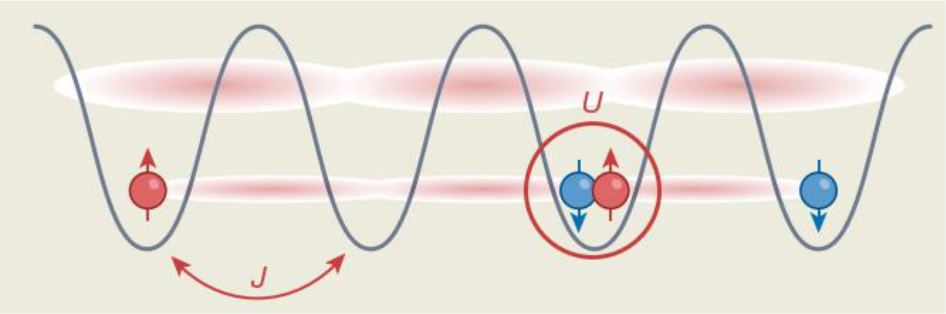
[2] Daley et al. (2022)

Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model

Hamiltonian

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



Real-time simulation [1]

Analog quantum simulator: ultracold atoms

Classical benchmark: tensor networks (MPS)

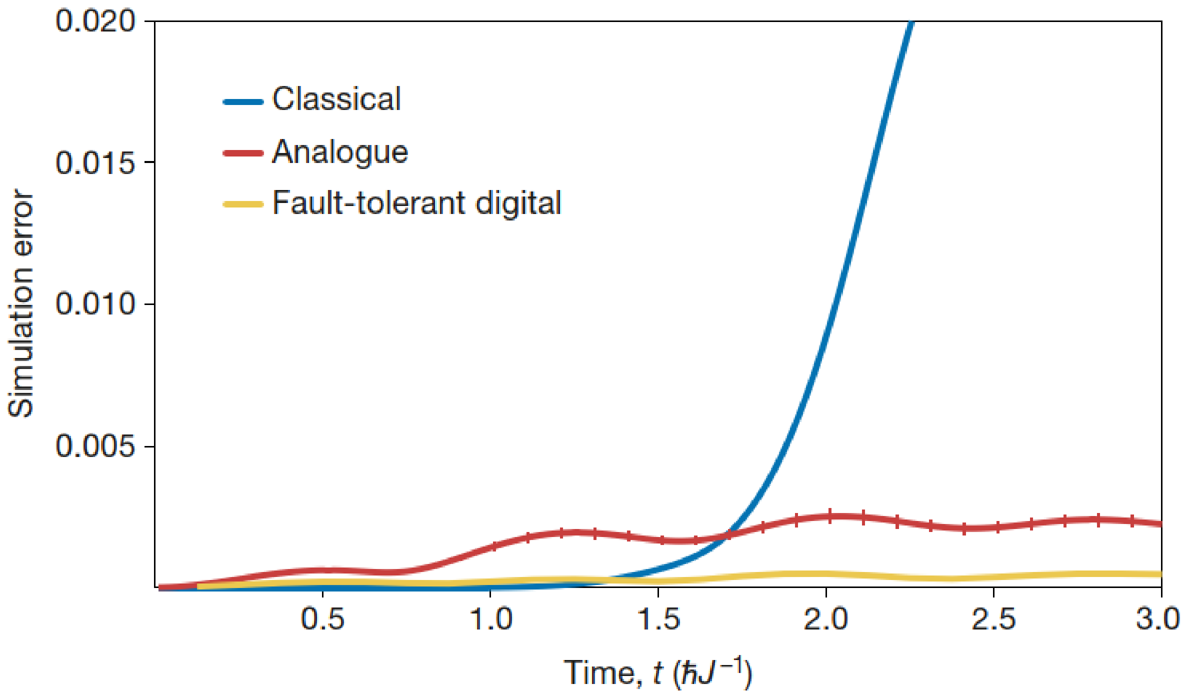
Experimental results

“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)

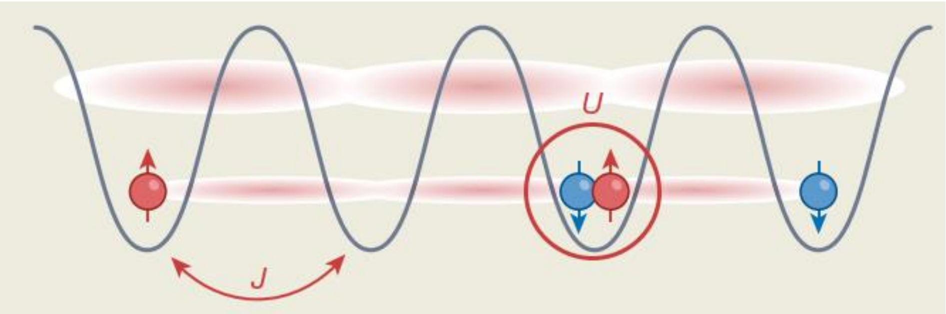
[2] Daley et al. (2022)

Do we really need *quantum* computing?

Example: 1+1D Bose-Hubbard model

Hamiltonian

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



Real-time simulation [1]

Analog quantum simulator: ultracold atoms

Classical benchmark: tensor networks (MPS)

Experimental results

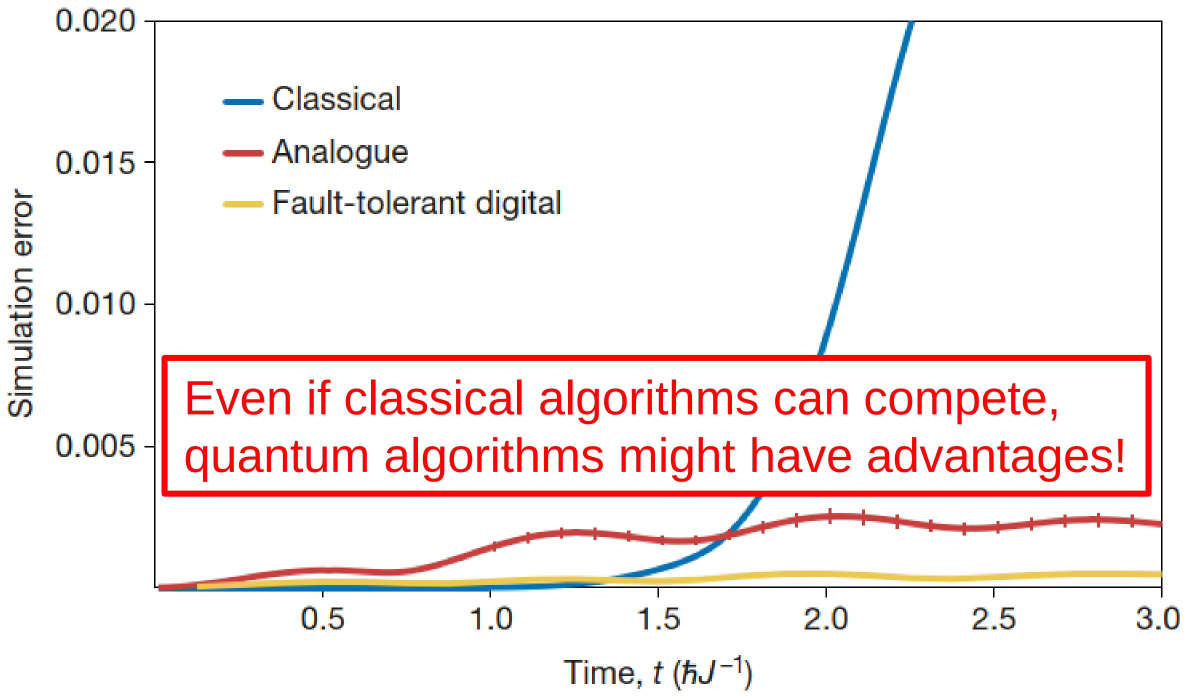
"the controlled [quantum] dynamics runs for longer times than present classical algorithms can keep track of" [1]

[1] Trotzky et al. (2012)

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



[2] Daley et al. (2022)

Quantum computing: where do we stand?

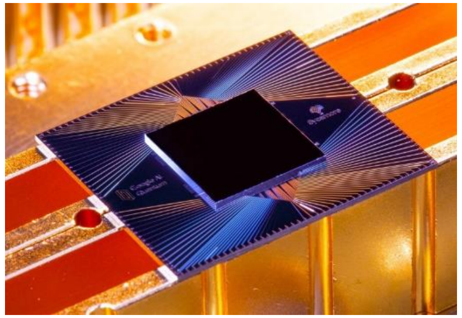
Quantum hardware	Quantum algorithms

Quantum computing: where do we stand?

Quantum hardware	Quantum algorithms

Achievements

Quantum advantage: outperformed classical computers [1]



Arute et al. (2019)

Zhong et al. (2020)

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

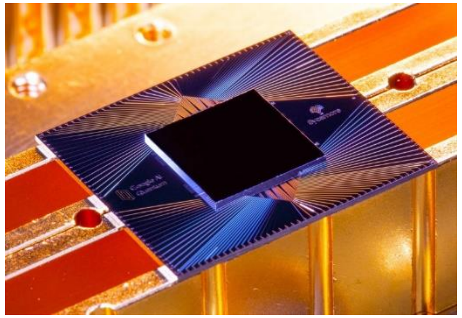
Quantum computing: where do we stand?

Quantum hardware	Quantum algorithms

Achievements

Quantum advantage: outperformed classical computers [1]

Exponential speedup of *specific* classical computations



Arute et al. (2019)

Zhong et al. (2020)

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

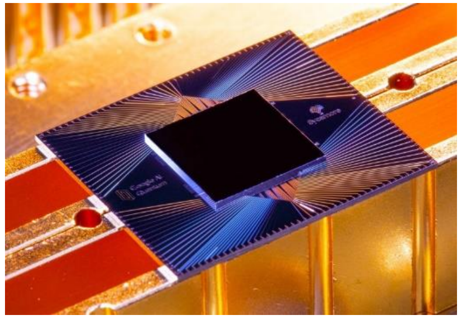
Quantum computing: where do we stand?

Quantum hardware	Quantum algorithms

Achievements

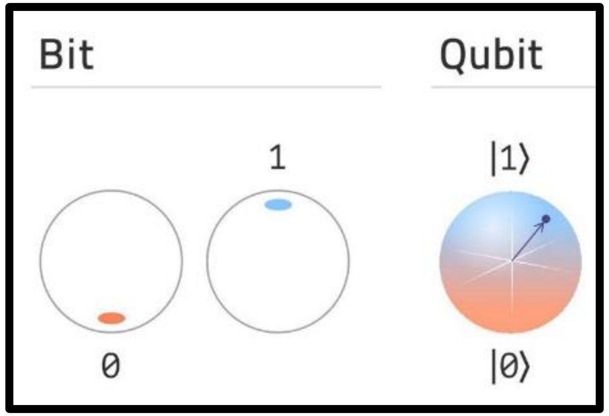
Quantum advantage: outperformed classical computers [1]

Exponential speedup of *specific* classical computations

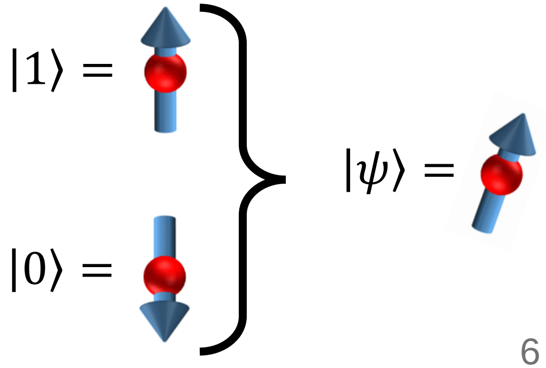


Arute et al. (2019)

Zhong et al. (2020)



Astibuag (2022)



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

Quantum computing: where do we stand?

Quantum hardware	**Quantum algorithms**

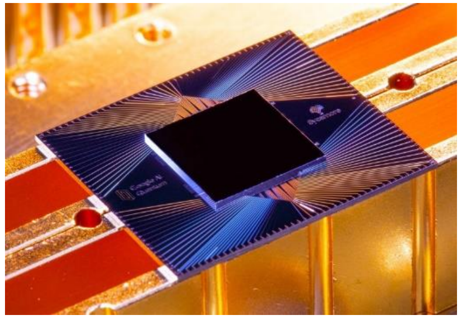
Achievements

Quantum advantage: outperformed classical computers [1]

Exponential speedup of *specific* classical computations

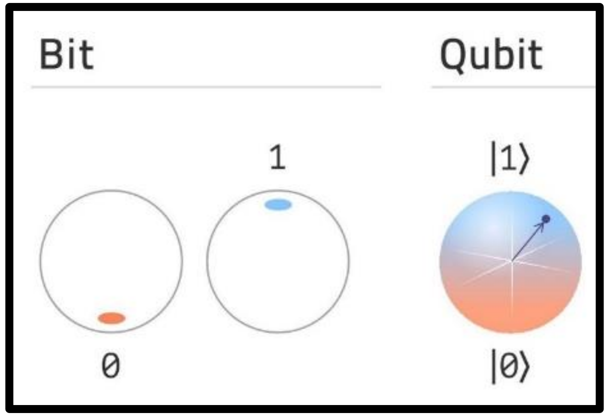
Challenges

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

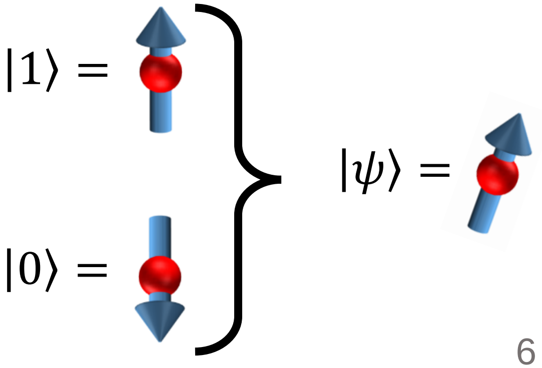


Arute et al. (2019)

Zhong et al. (2020)



Astibuag (2022)



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

Quantum computing: where do we stand?

Quantum hardware	**Quantum algorithms**

Achievements

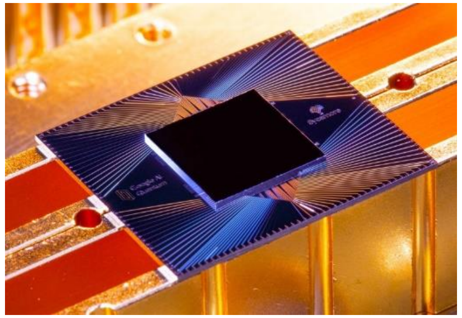
Quantum advantage: outperformed classical computers [1]

Exponential speedup of *specific* classical computations

Challenges

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

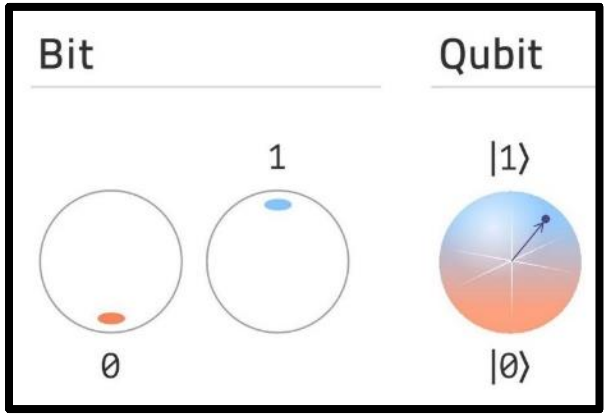
noise → need quantum error mitigation / correction



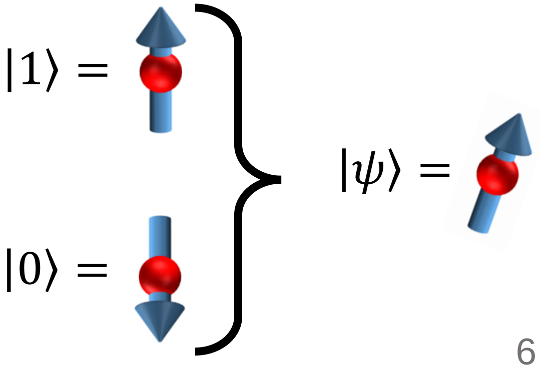
Arute et al. (2019)

Zhong et al. (2020)

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



Astibuag (2022)



Quantum computing: where do we stand?

Quantum hardware	Quantum algorithms

Achievements

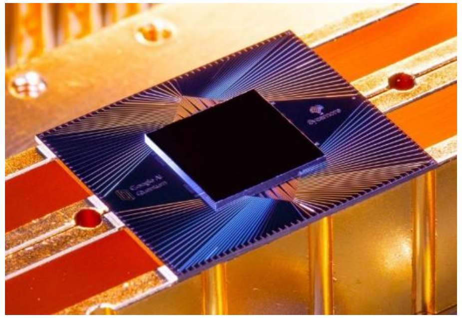
Quantum advantage: outperformed classical computers [1]

Exponential speedup of *specific* classical computations

Challenges

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

noise → need quantum error mitigation / correction



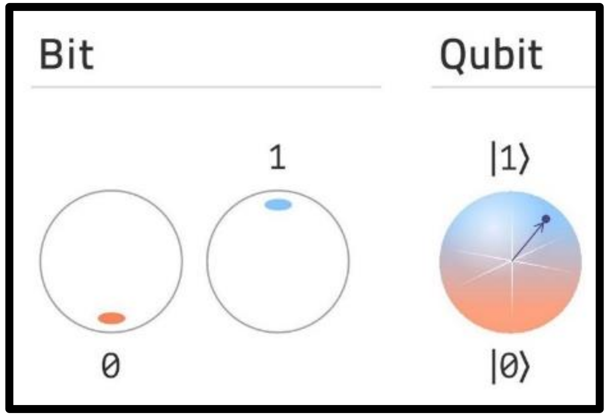
Arute et al. (2019)

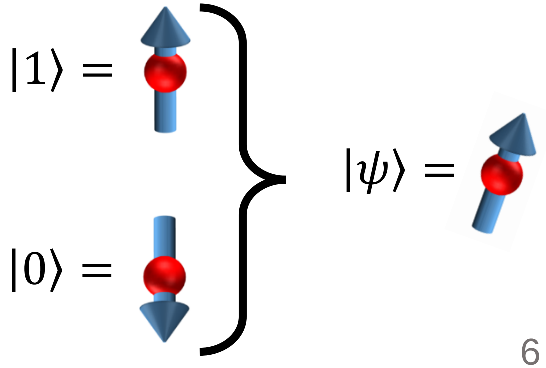
Zhong et al. (2020)

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

Applications

cryptography, optimization problems, …





Astibuag (2022)

Quantum computing: where do we stand?

Quantum hardware

Achievements

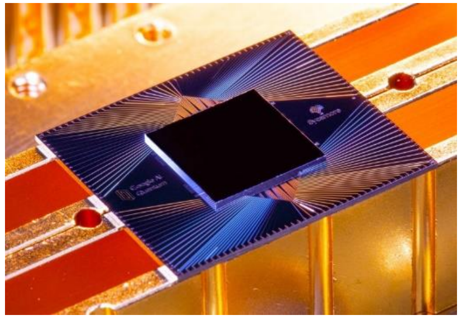
Quantum advantage: outperformed classical computers [1]

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)

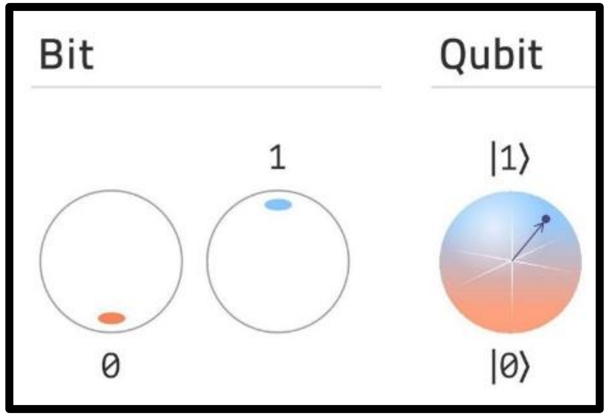
[1] Arute et al. (2019), Zhong et al. (2020, 2021), Yulin (2021), Zhu (2021), Madsen (2022).

Quantum algorithms

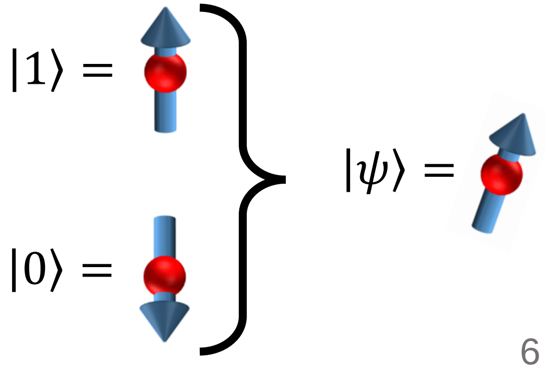
Applications

cryptography, optimization problems, …

particle / nuclear / condensed matter physics, …



Astibuag (2022)



Quantum computing: where do we stand?

Quantum hardware

Achievements

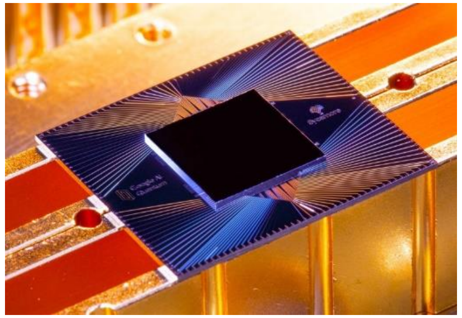
Quantum advantage: outperformed classical computers [1]

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).

Quantum algorithms

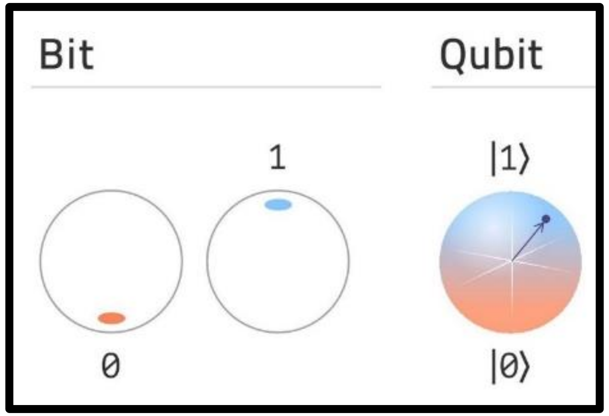
Applications

cryptography, optimization problems, …

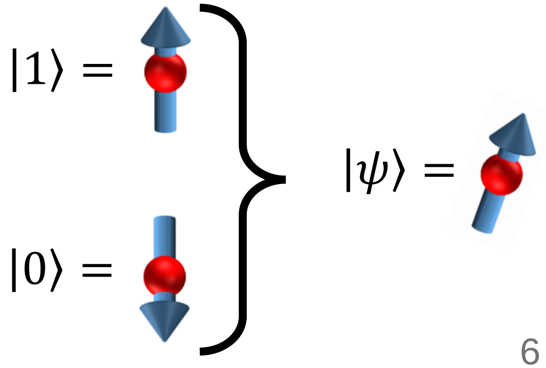
particle / nuclear / condensed matter physics, …

Challenges

new technology → need fundamentally new algorithms



Astibuag (2022)



Quantum computing: where do we stand?

Quantum hardware	Quantum algorithms

Achievements

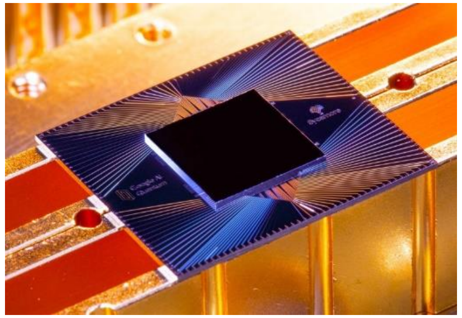
Quantum advantage: outperformed classical computers [1]

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

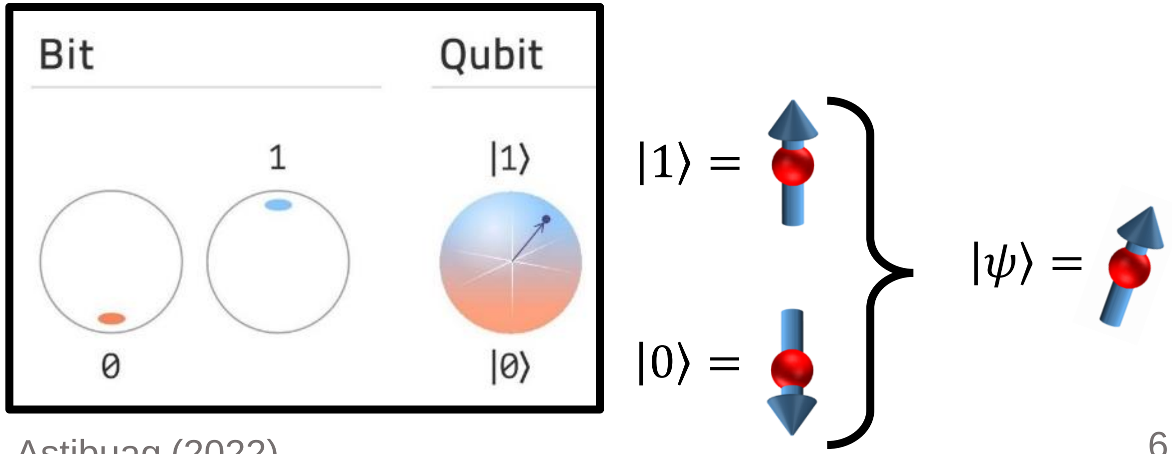
cryptography, optimization problems, …

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)	Example: “classical advantage” (2021)

Quantum computing: where do we stand?

Example: “quantum advantage” (2019)	Example: “classical advantage” (2021)

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 computing: where do we stand?

Example: "quantum advantage" (2019)	Example: "classical advantage" (2021)

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.

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)$\times(1+40+1)$(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.

Quantum computing: where do we stand?

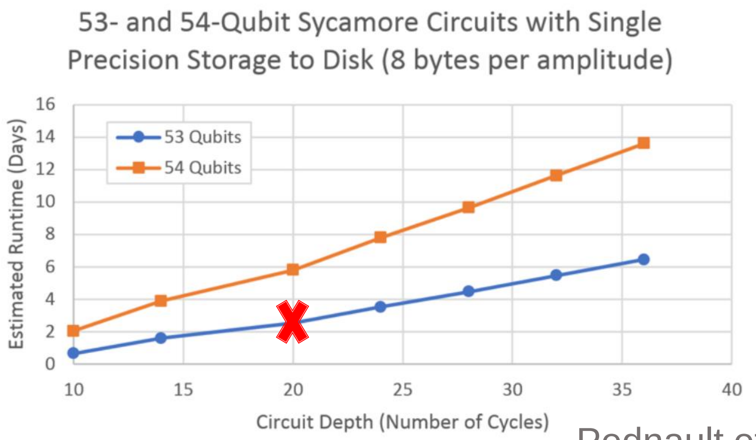
Example: “quantum advantage” (2019)	**Example: “classical advantage” (2021)**

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.

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 do we stand?

Example: “quantum advantage” (2019)

Example: “classical advantage” (2021)

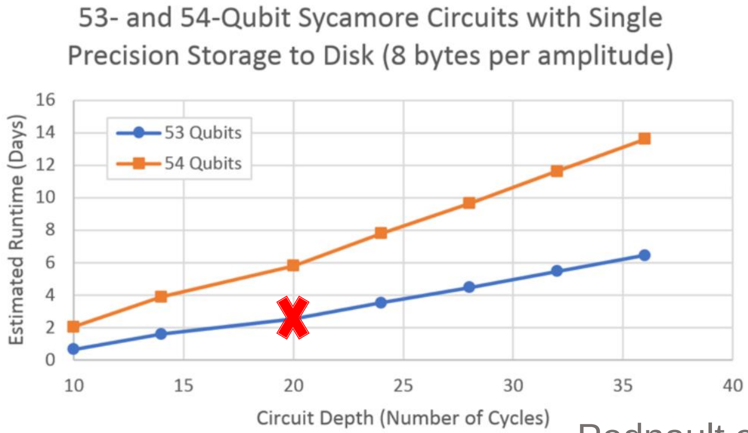
Quantum supremacy using a programmable superconducting processor

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

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)$\times(1+40+1)$(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 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.

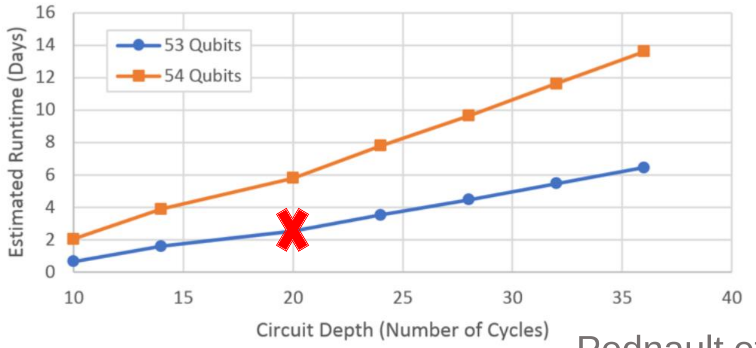
→ **Quantum-classical race:**
algorithms and hardware quickly advance
→ **For exponentially hard problems:**
small quantum step ↔ giant classical leap

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.

53- and 54-Qubit Sycamore Circuits with Single Precision Storage to Disk (8 bytes per amplitude)



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

Pednault et al. (2019)

Quantum computing: where will we go?

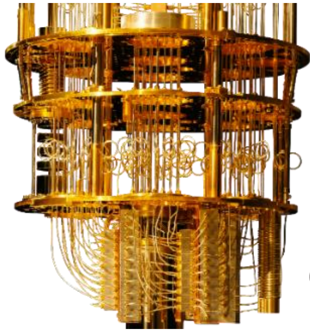
The Path to Go…	A Rough Sketch…

Quantum computing: where will we go?

The Path to Go…	A Rough Sketch…

State of the Art

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



Carlow (2018)

Quantum computing: where will we go?

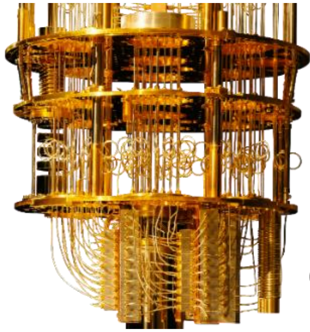
The Path to Go…	A Rough Sketch…

State of the Art

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

Near Future

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



Carlow (2018)

Quantum computing: where will we go?

The Path to Go…	A Rough Sketch…

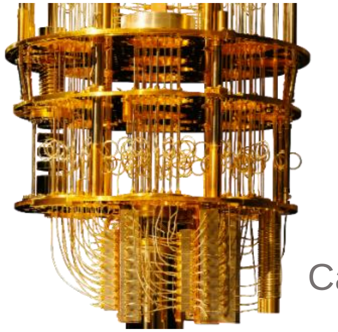
State of the Art

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

Near Future

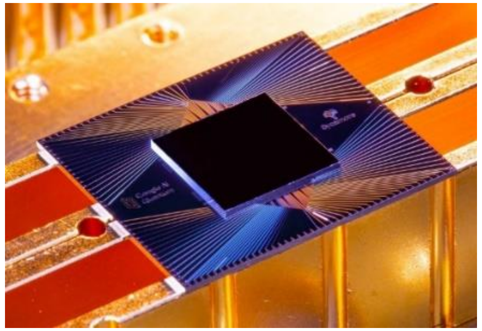
IBM: 433 (2022) → 1121 (2023) → 4158 (2025) → …

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



Carlow (2018)

Arute et al. (2019)



Quantum computing: where will we go?

The Path to Go…	A Rough Sketch…

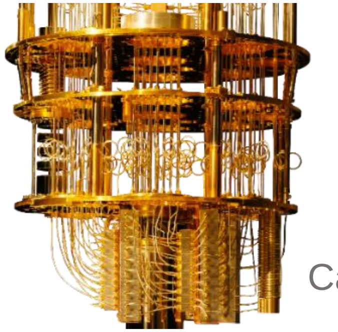
State of the Art

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

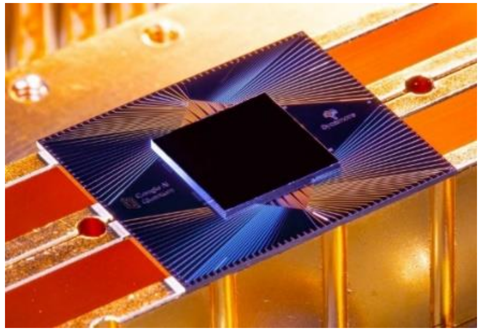
Near Future

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

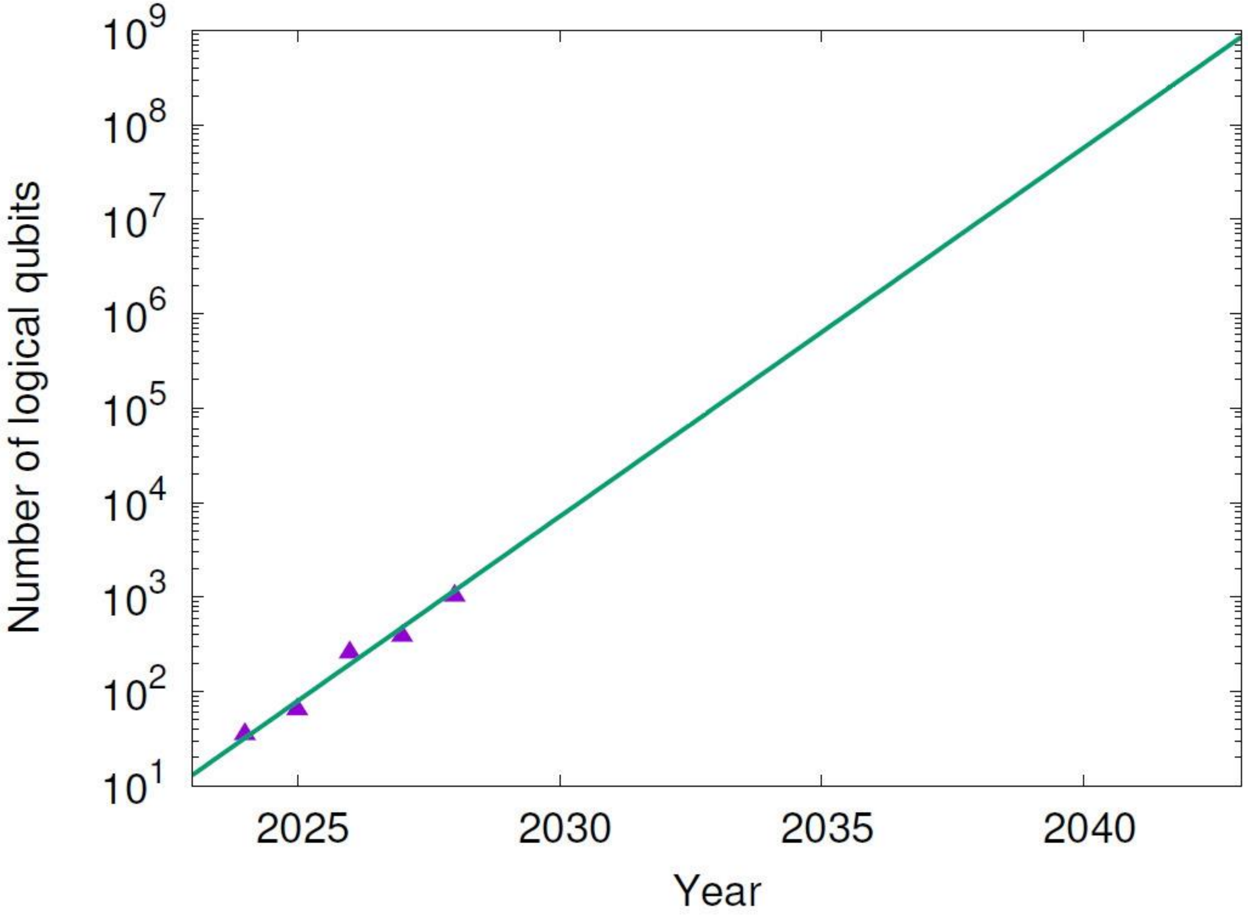
Google: 1,000,000 physical / 1000 logical qubits (2029)?



Carlow (2018)



Arute et al. (2019)



Kan, Nam (2021)

Quantum computing: where will we go?

The Path to Go…	A Rough Sketch…

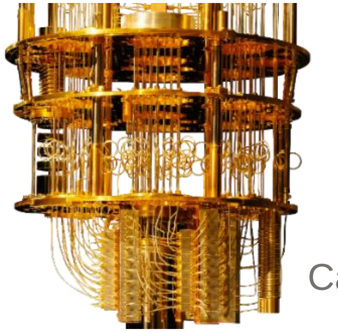
State of the Art

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

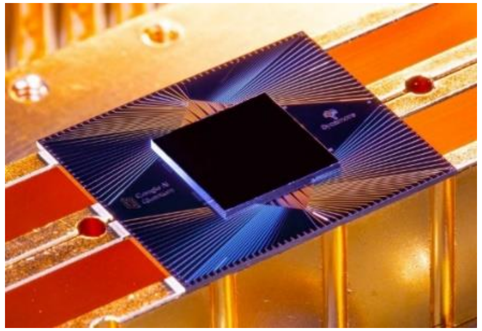
Near Future

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

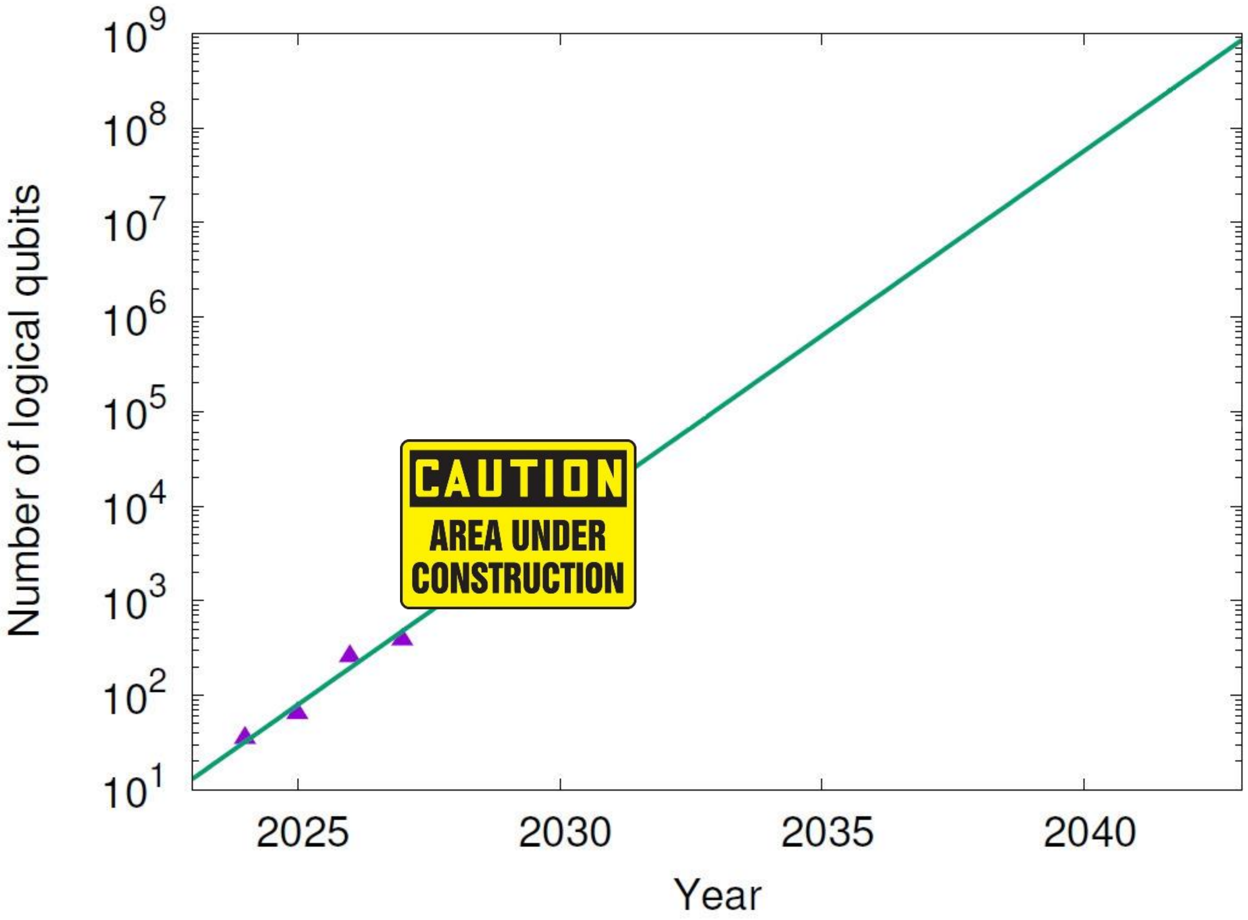
Google: 1,000,000 physical / 1000 logical qubits (2029)?



Carlow (2018)



Arute et al. (2019)



Kan, Nam (2021)

Quantum computing: where will we go?

The Path to Go…	A Rough Sketch…

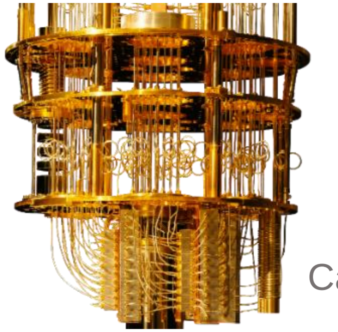
State of the Art

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

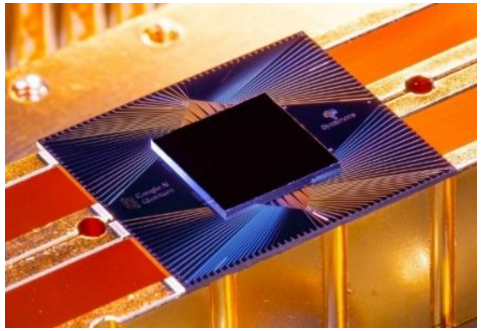
Near Future

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

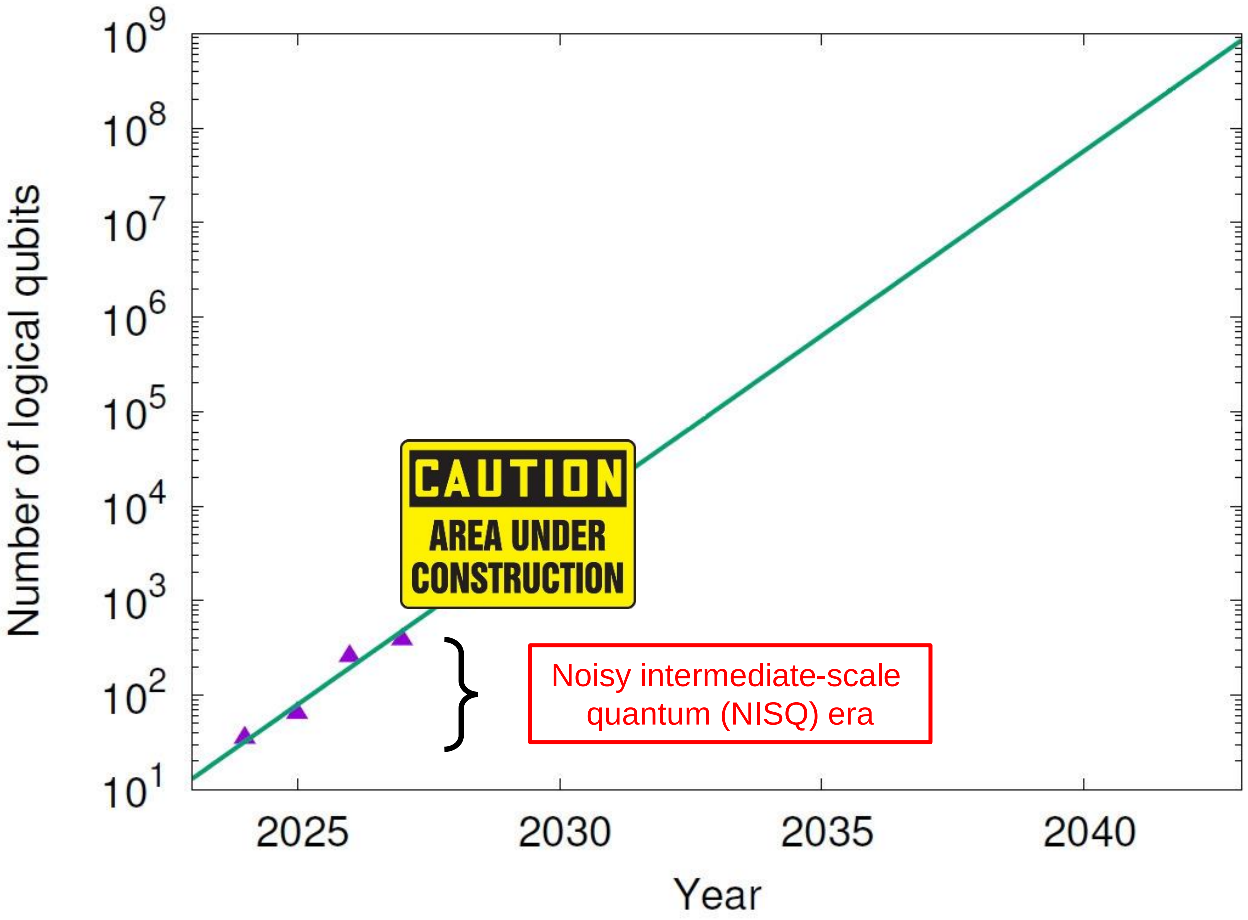
Google: 1,000,000 physical / 1000 logical qubits (2029)?



Carlow (2018)



Arute et al. (2019)



Kan, Nam (2021)

Quantum computing: where will we go?

The Path to Go…

A Rough Sketch…

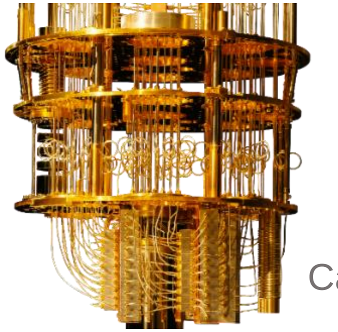
State of the Art

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

Near Future

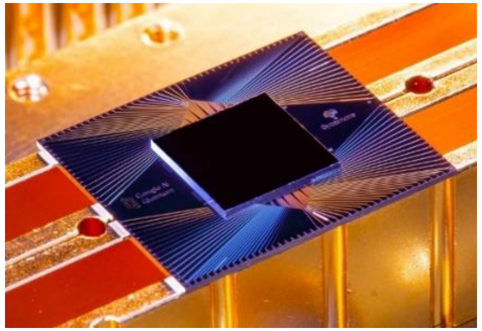
IBM: 433 (2022) → 1121 (2023) → 4158 (2025) → …

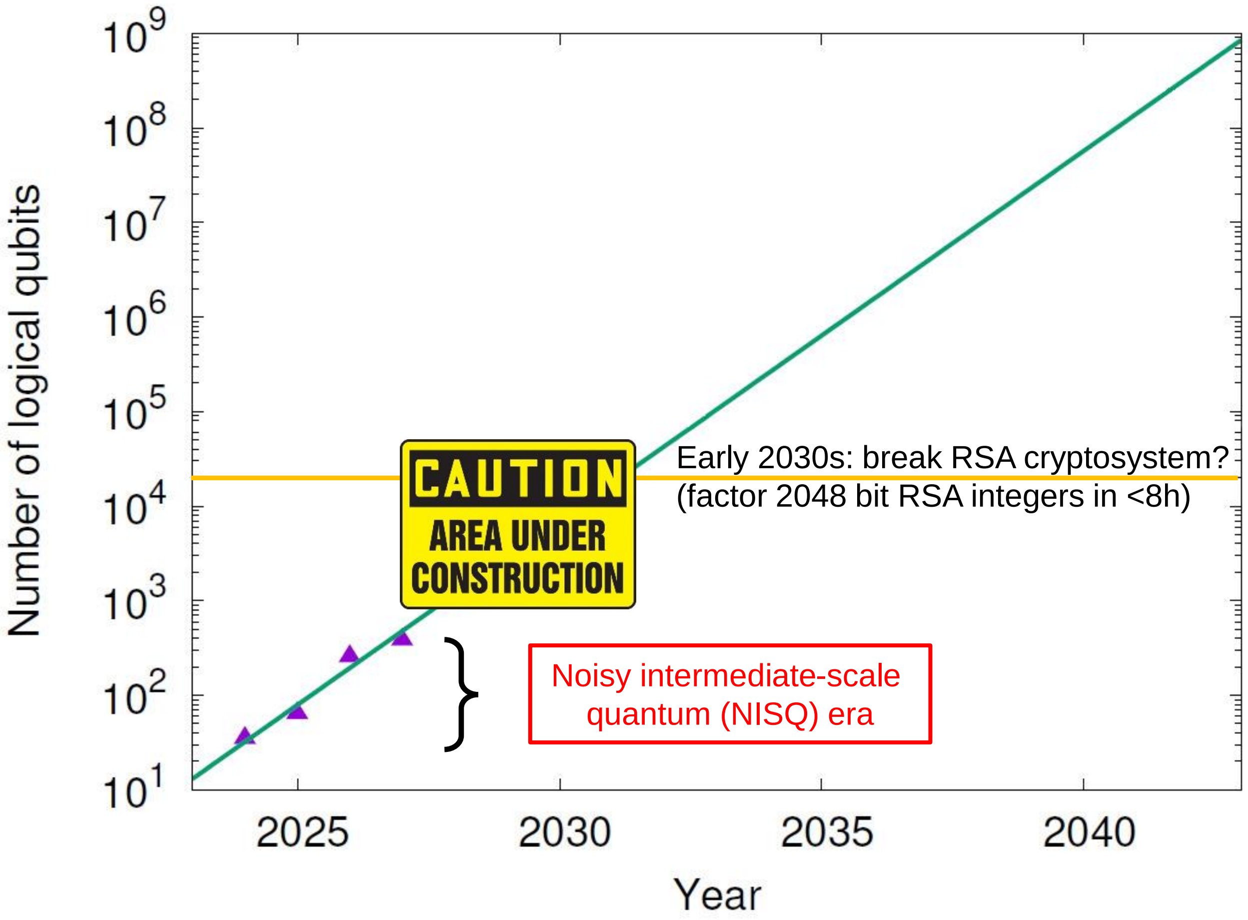
Google: 1,000,000 physical / 1000 logical qubits (2029)?



Carlow (2018)

Arute et al. (2019)





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

Quantum computing: where will we go?

The Path to Go… | **A Rough Sketch…**

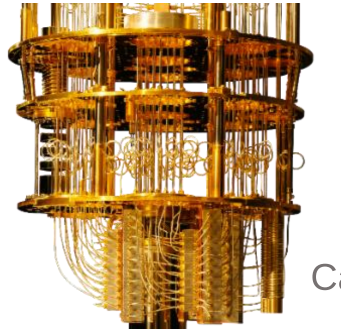
State of the Art

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

Near Future

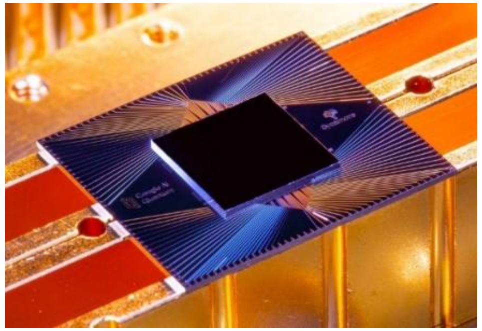
IBM: 433 (2022) → 1121 (2023) → 4158 (2025) → …

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



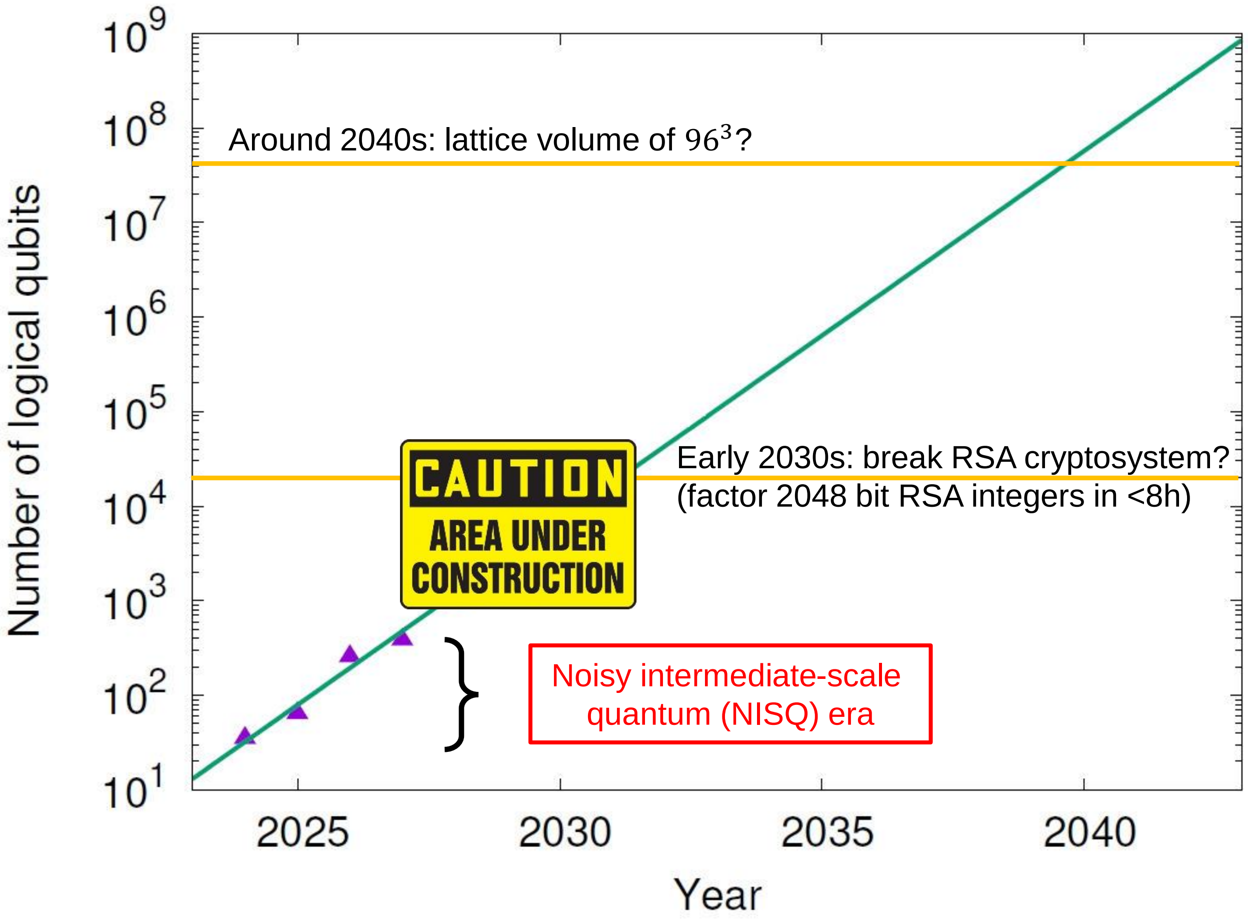
Carlow (2018)

Arute et al. (2019)



Far Future

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



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

Quantum computing: where will we go?

The Path to Go…

A Rough Sketch…

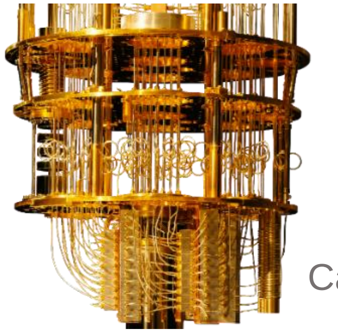
State of the Art

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

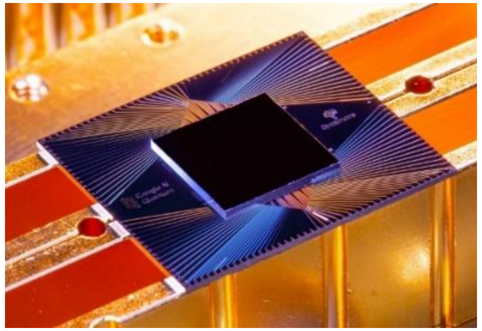
Near Future

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

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



Carlow (2018)

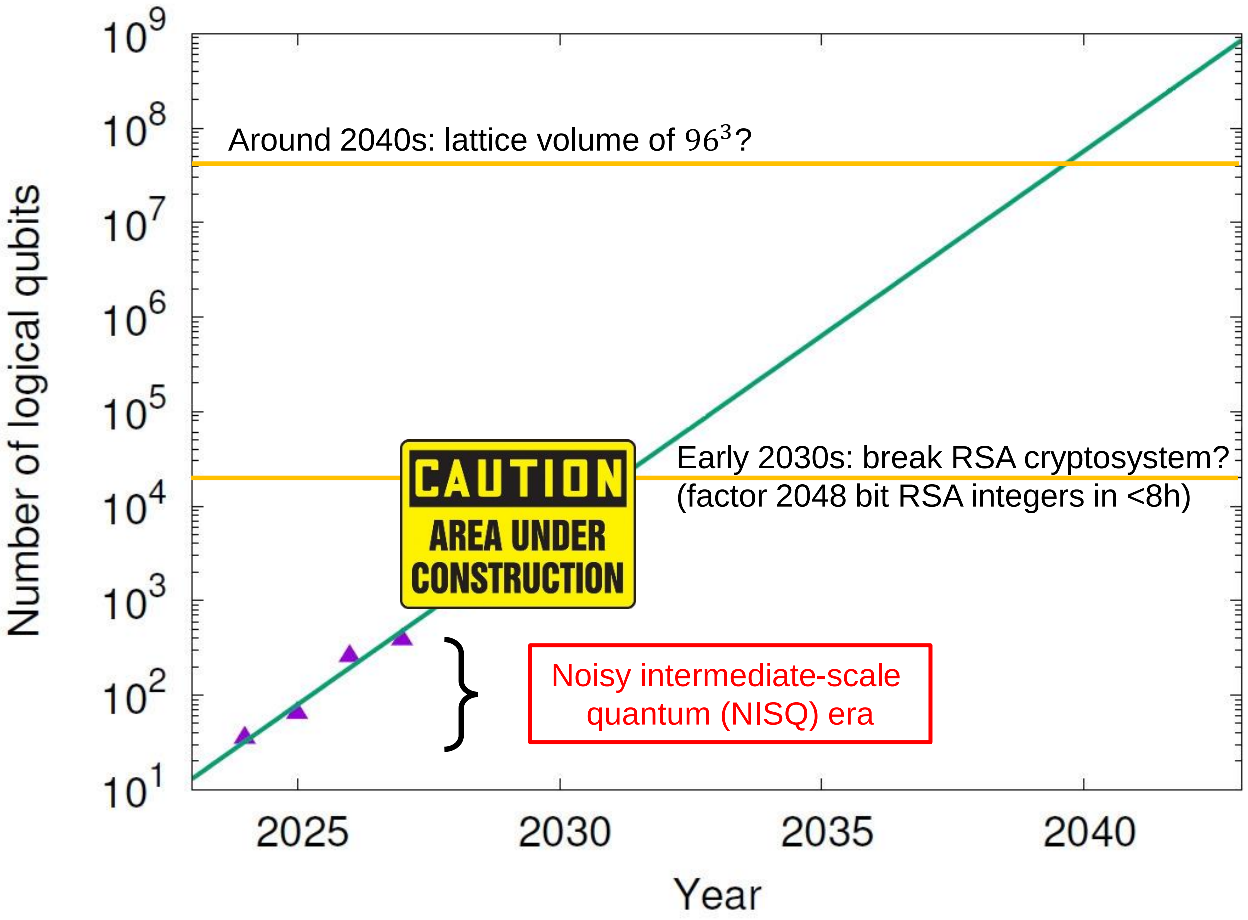


Arute et al. (2019)

Far Future

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

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



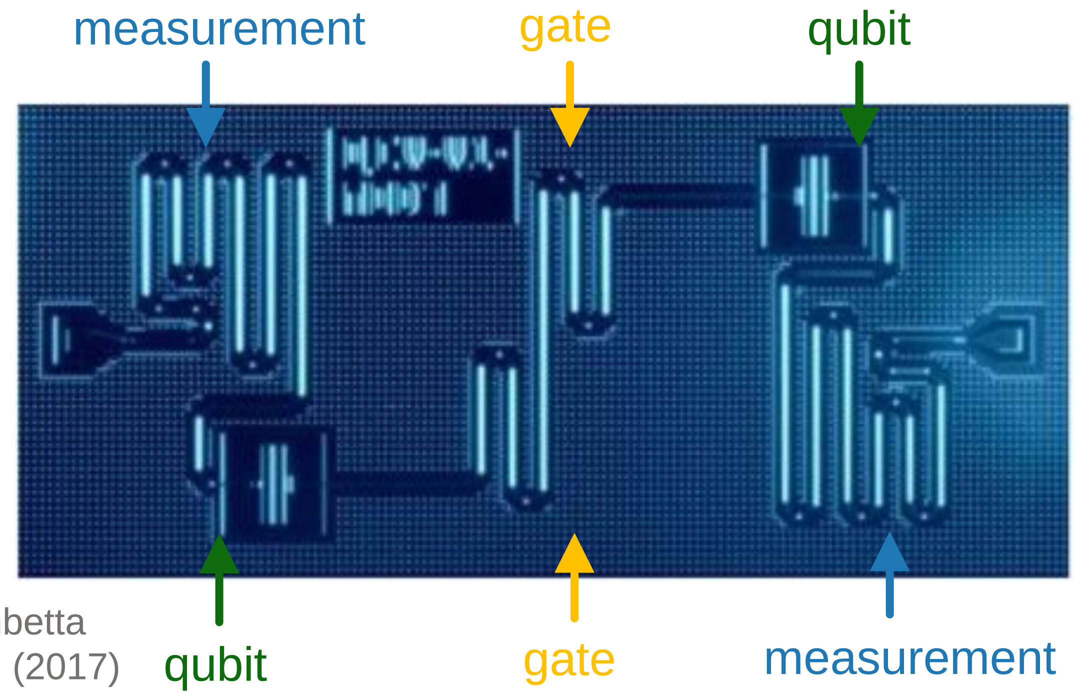
Gidney, Ekera (2019);
Kan, Nam (2021)

How can we reduce the noise?

Noisy quantum circuit	Error mitigation versus error correction

How can we reduce the noise?

Noisy quantum circuit	Error mitigation versus error correction

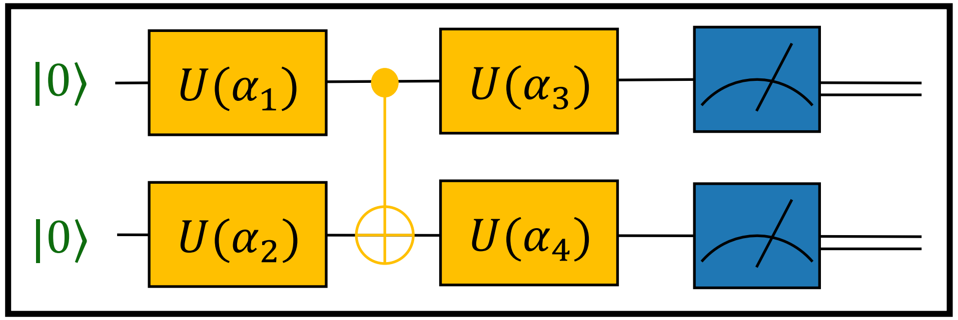


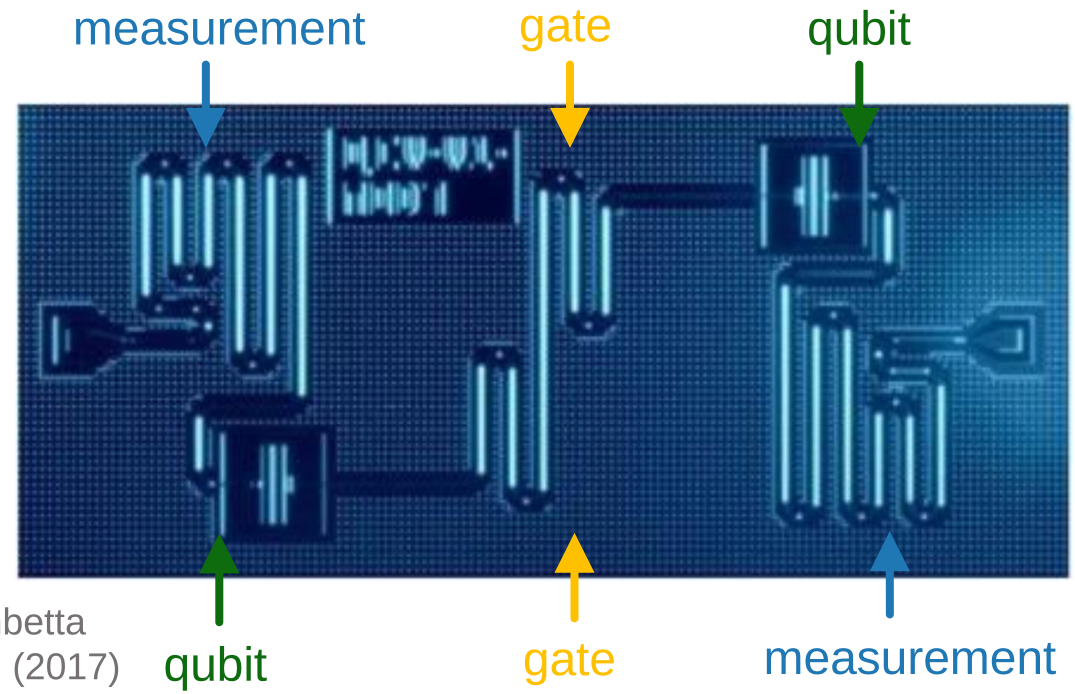
Gambetta et al. (2017)

How can we reduce the noise?

Noisy quantum circuit	Error mitigation versus error correction

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

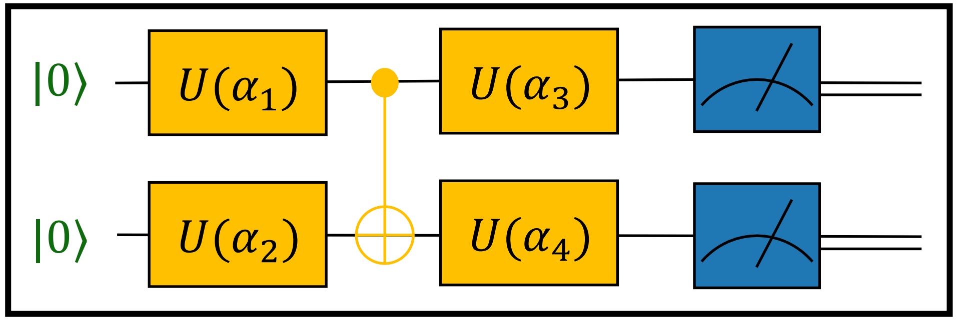


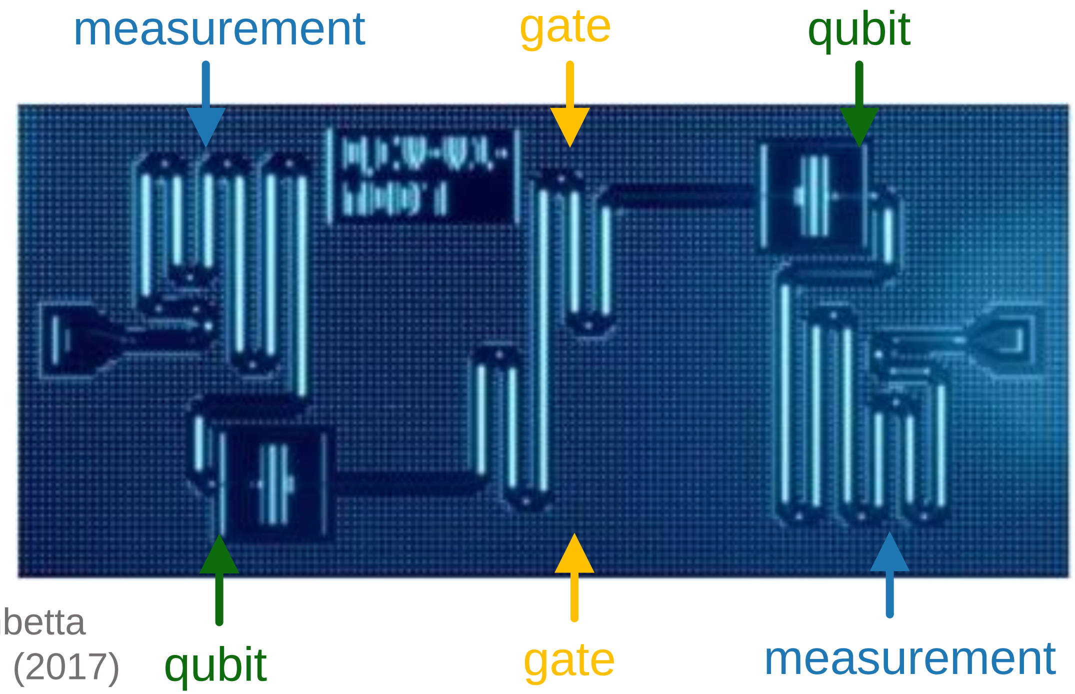


Gambetta
et al. (2017)

How can we reduce the noise?

Noisy quantum circuit

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



Gambetta et al. (2017)

Error mitigation versus error correction

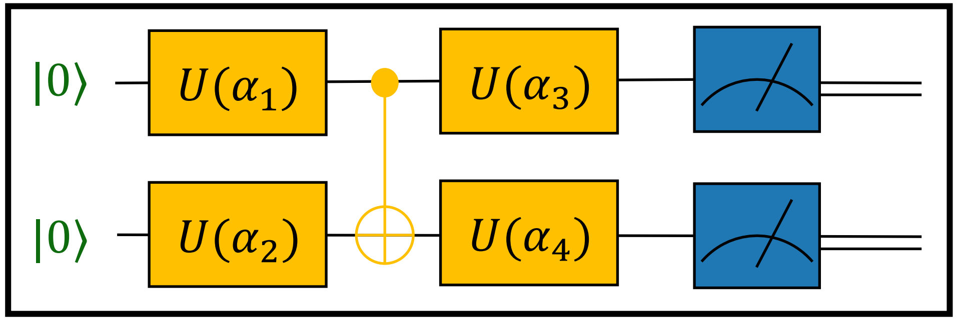
Problem

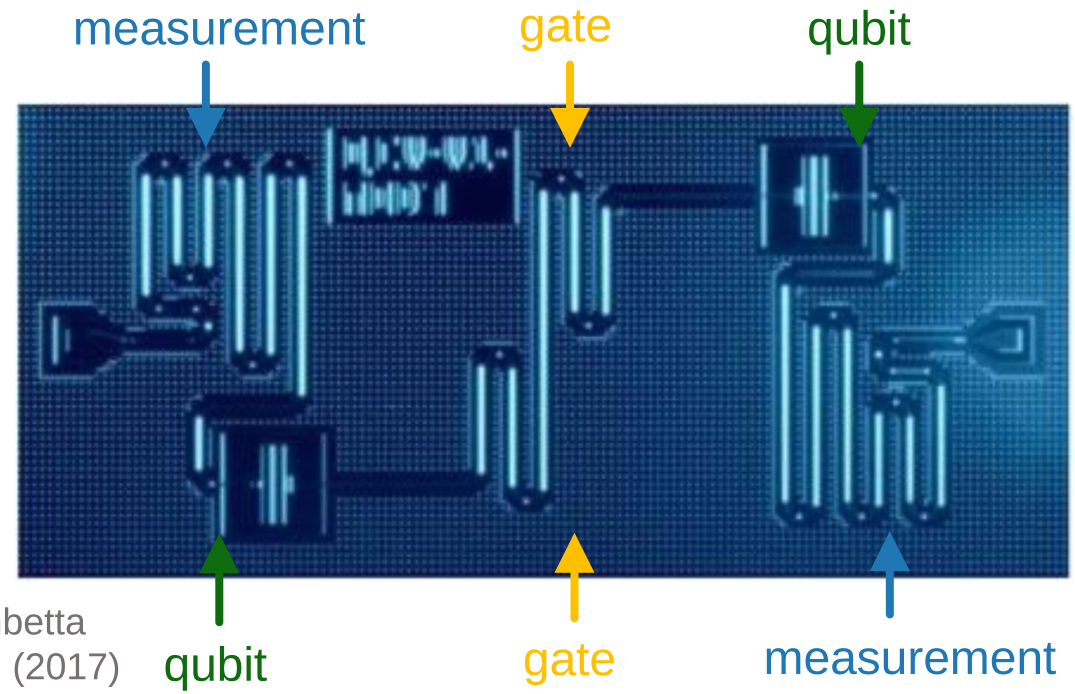
Quantum noise: affecting qubits, gates, measurement

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

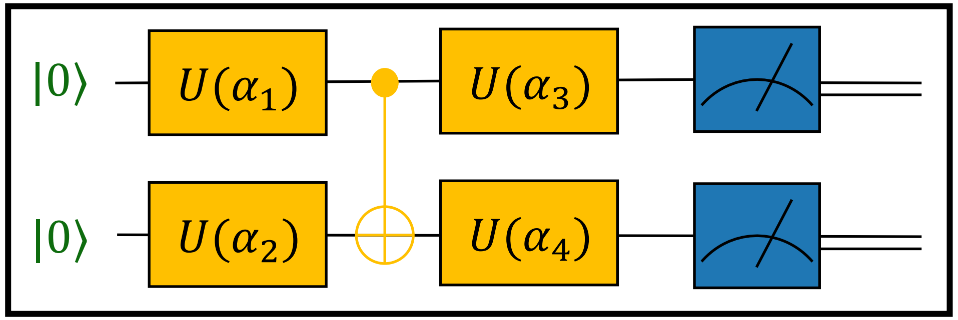
Near-term solution

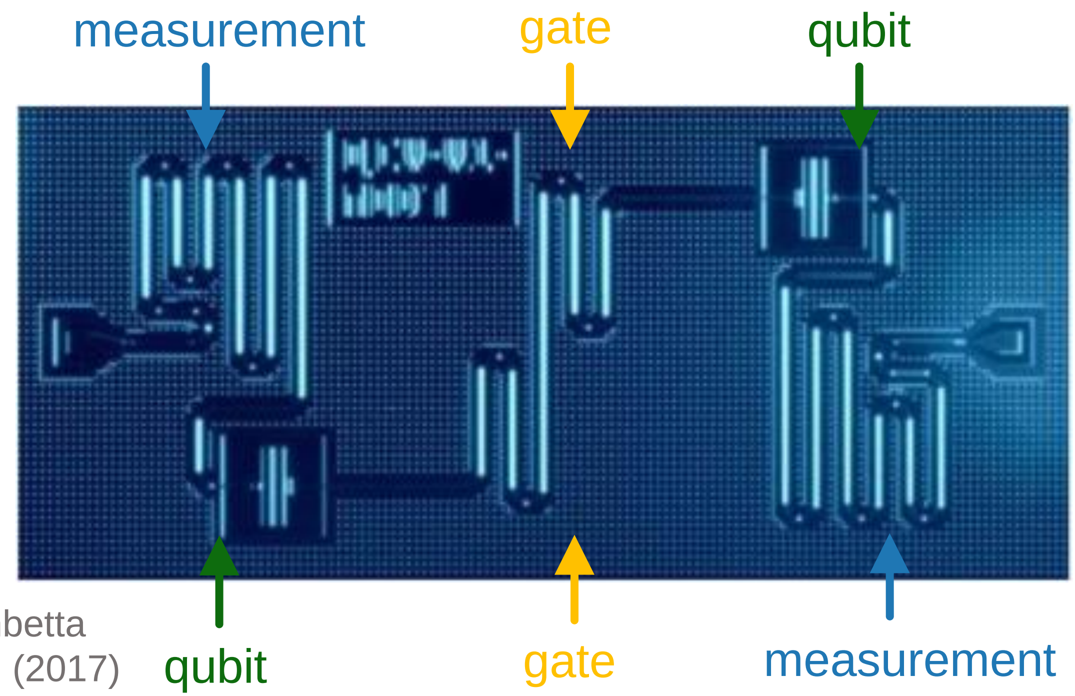
Error mitigation: reduce noise on NISQ devices

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

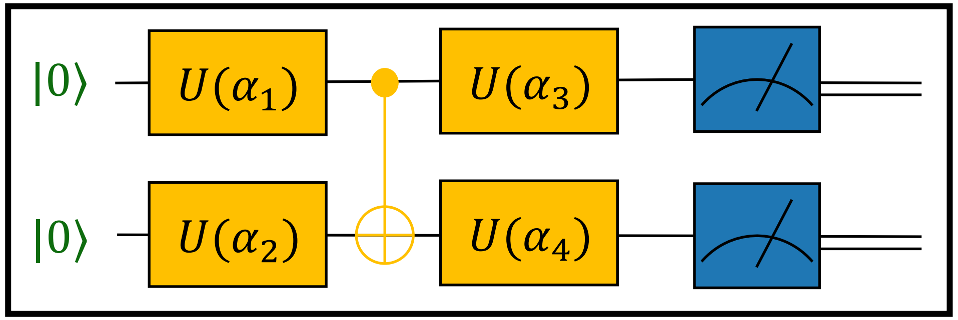
Long-term solution

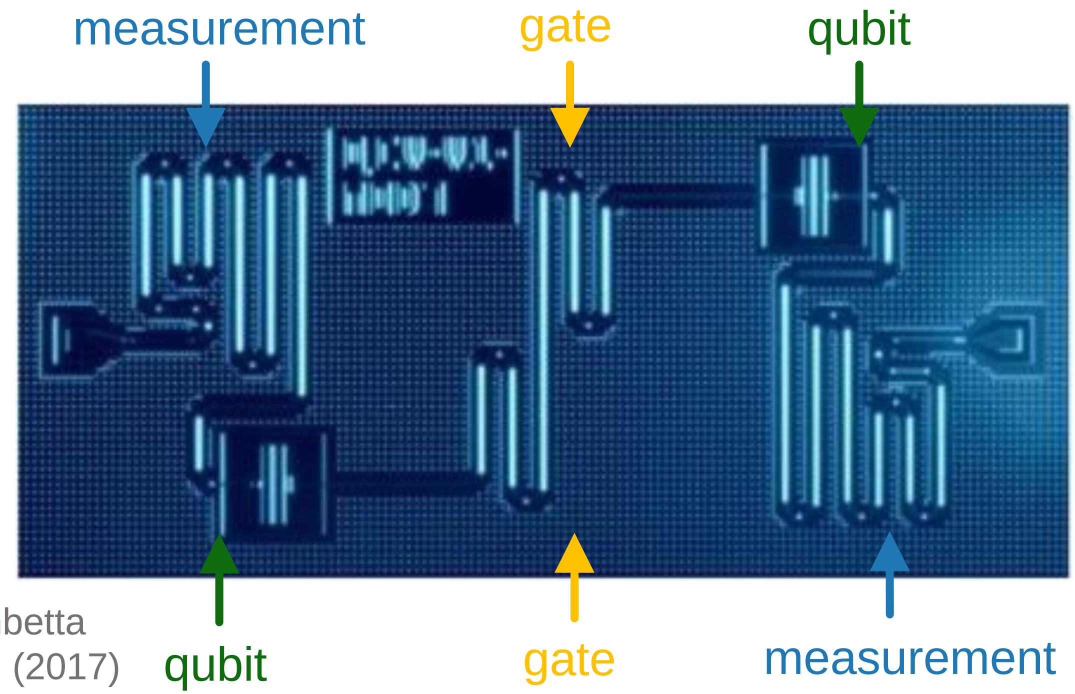
Error correction (EC): fault-tolerant quantum computation

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

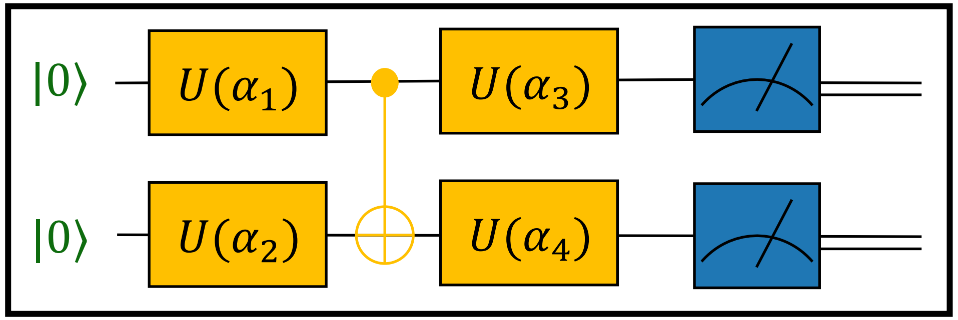
E.g. bit-flip code,[1] Shor code,[2] toric code,[3] GKP code,[4] …

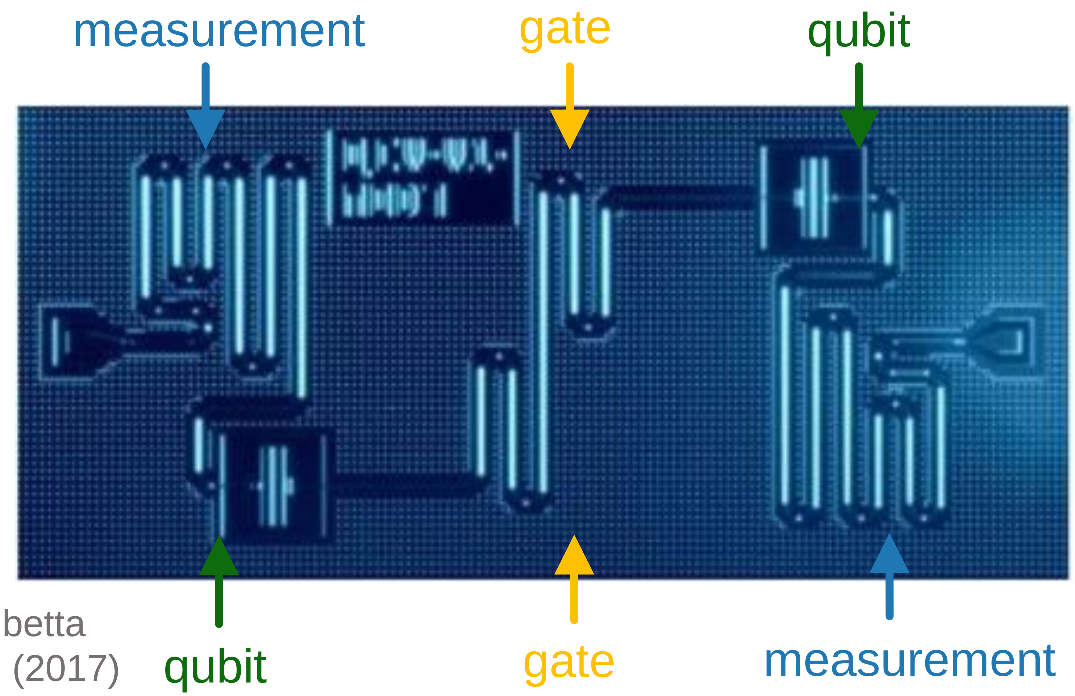
[1] Peres (1985), [2] Shor (1995), [3] Kitaev (1997), [4] Gottesmann et al. (2001), ...

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

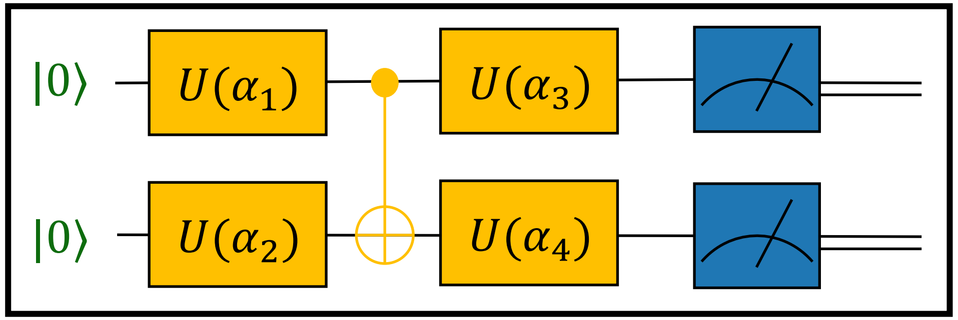
For EC, need extra qubits and noise below *threshold*[5]

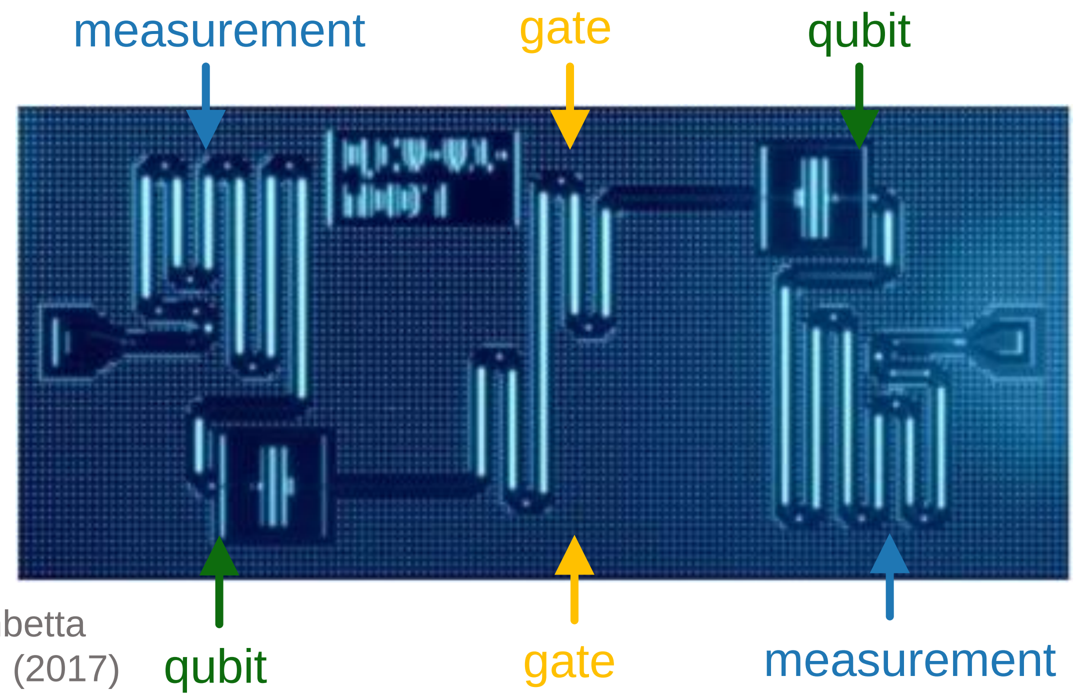
[1] Peres (1985), [2] Shor (1995), [3] Kitaev (1997), [4] Gottesmann et al. (2001), ...
[5] Shor (1996), Knill et al. (1998), Kitaev (2003), Aharonov et al. (2008)

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. surface 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)

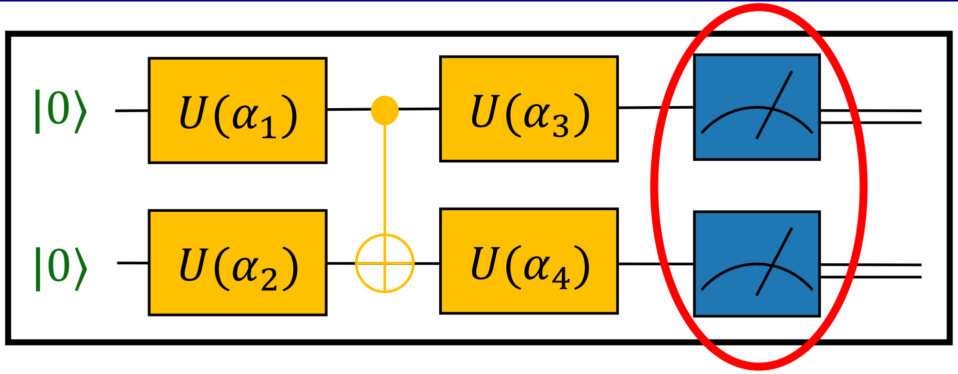
Error mitigation: how can we reduce the noise?

Example: measurement error mitigation	Operator rescaling method

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

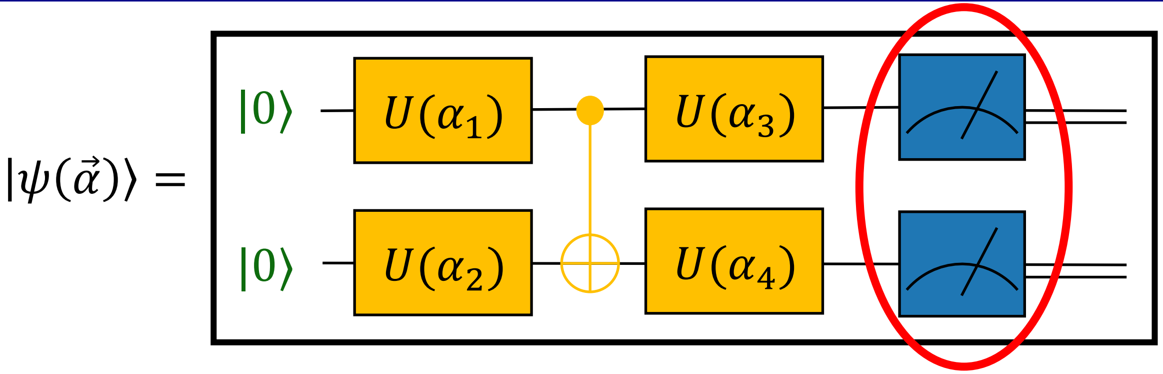
$|\psi(\vec{\alpha})\rangle =$



Operator rescaling method

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation



Operator rescaling method

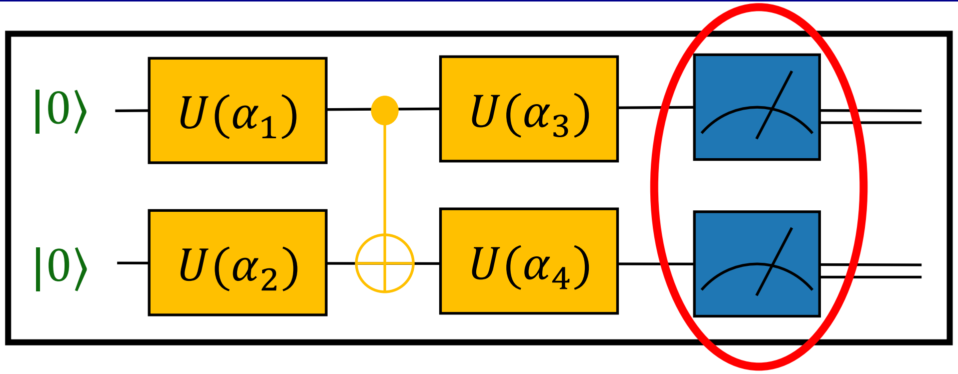
Goal

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

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

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



Operator rescaling method

Goal

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

Method [1]

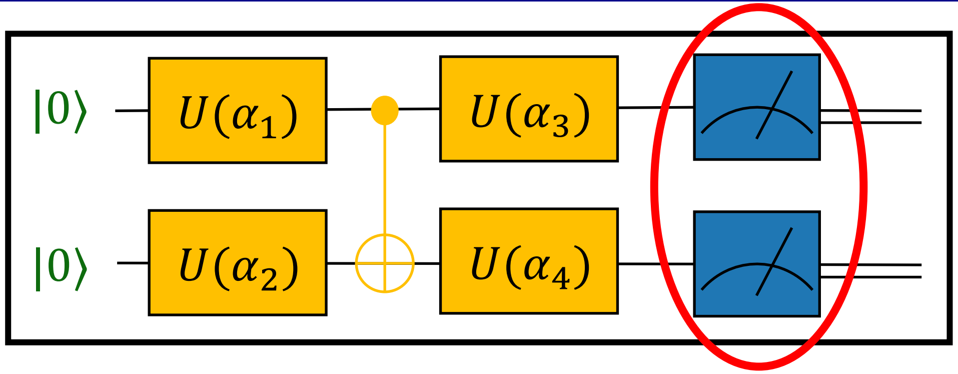
replace operators by noisy operators: $\langle\tilde{\psi}|\boldsymbol{O}|\tilde{\psi}\rangle \rightarrow \langle\psi|\tilde{O}|\psi\rangle$

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

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

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



Operator rescaling method

Goal

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

Method [1]

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

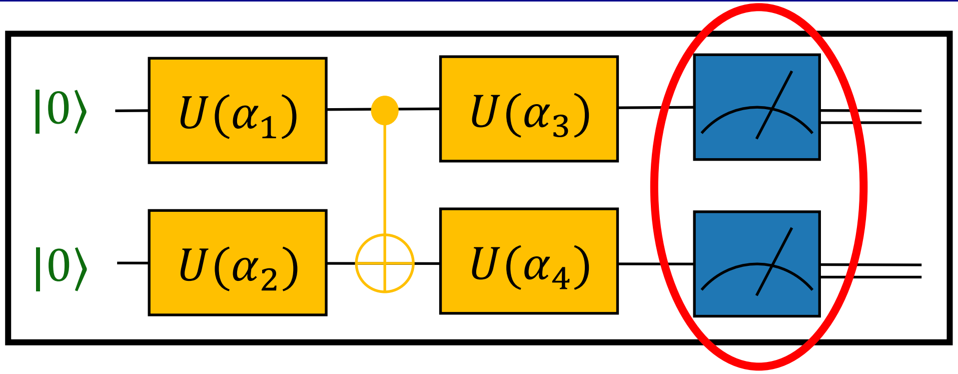
Readout	Bit Flips	Probability	Noisy Operator
correct	$0 \rightarrow 0,\ 1 \rightarrow 1$	$(1-p_0)(1-p_1)$	$\tilde{O} = \boldsymbol{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

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

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

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



Operator rescaling method

Goal

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

Method [1]

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

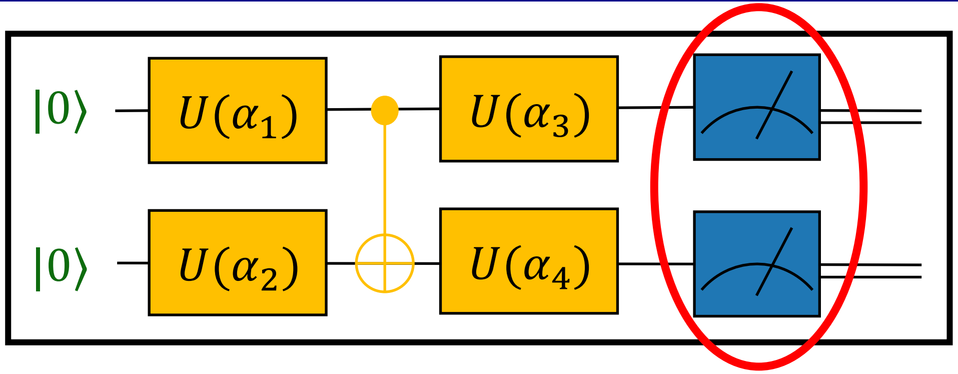
Readout	Bit Flips	Probability	Noisy Operator
correct	$0 \rightarrow 0,\ 1 \rightarrow 1$	$(1-p_0)(1-p_1)$	$\tilde{O} = \boldsymbol{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
incorrect … for both outcomes	$0 \rightarrow 1,\ 1 \rightarrow 0$	$p_0 p_1$	$\tilde{O} = -Z = \begin{pmatrix} -1 & 0 \\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

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

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

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



Operator rescaling method

Goal

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

Method [1]

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

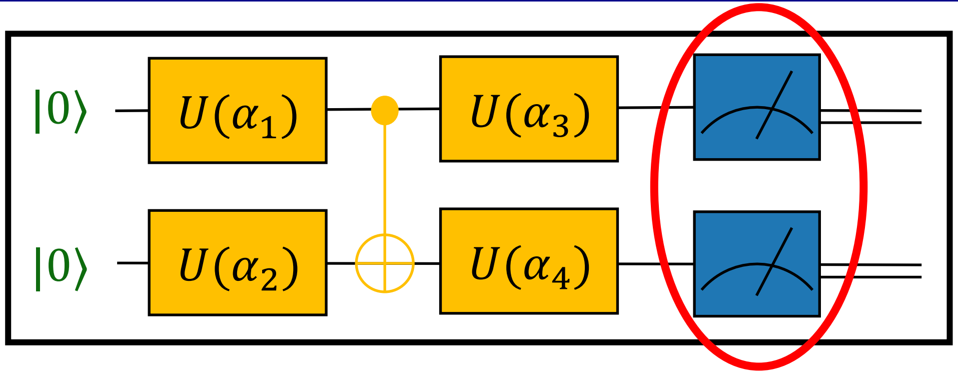
Readout	Bit Flips	Probability	Noisy Operator
correct	$0 \rightarrow 0,\ 1 \rightarrow 1$	$(1-p_0)(1-p_1)$	$\tilde{O} = \boldsymbol{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
incorrect … for both outcomes	$0 \rightarrow 1,\ 1 \rightarrow 0$	$p_0 p_1$	$\tilde{O} = -Z = \begin{pmatrix} -1 & 0 \\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

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

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

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



Operator rescaling method

Goal

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

Method [1]

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

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	$0 \to 0,\ 1 \to 0$	$(1-p_0)p_1$	$\tilde{O} = \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Total noisy operator: $\tilde{O}$

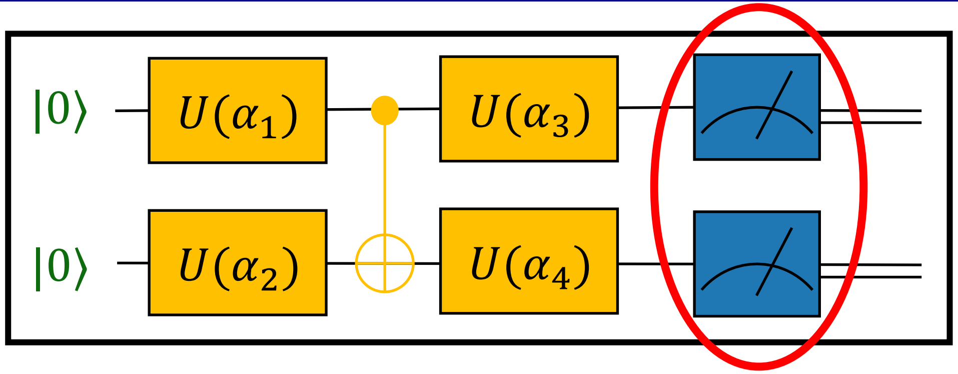
$$= (1-p_0)(1-p_1)\boldsymbol{Z} + p_0 p_1(-Z) + p_0(1-p_1)(-\mathbb{I}) + (1-p_0)p_1\mathbb{I}$$

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

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

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



Operator rescaling method

Goal

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

Method [1]

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

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

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:

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

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

Error mitigation: how can we reduce the noise?

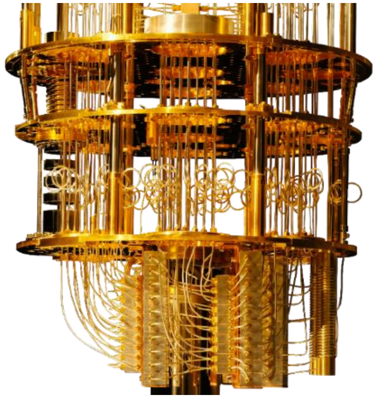
Example: measurement error mitigation	Example: gate error mitigation

Error mitigation: how can we reduce the noise?

Example: measurement error mitigation	Example: gate error mitigation

Operator rescaling method[1]

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



Carlow (2018)

[1] Kandala et al. (2017), Yeter-Aydeniz et al. (2019), LF et al. (2020)

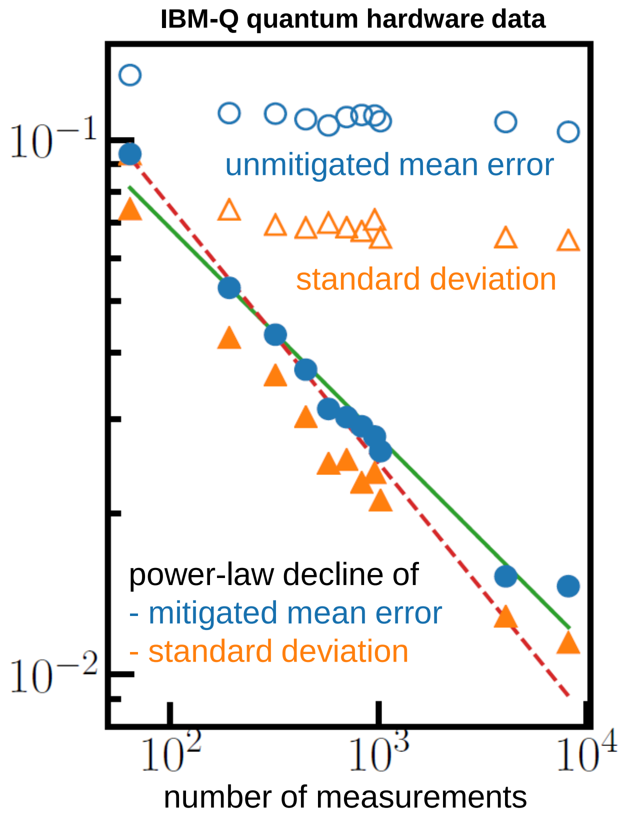
Error mitigation: how can we reduce the noise?

Example: measurement error mitigation	Example: gate error mitigation

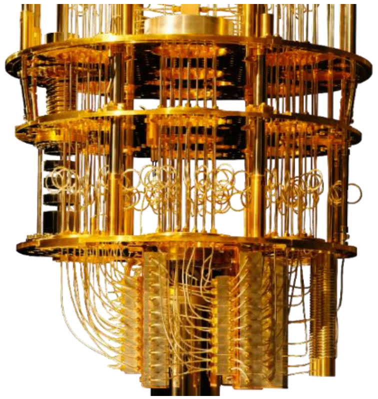
Operator rescaling method[1]

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

Result: measurement error reduced by factor 10



LF et al. (2020)



Carlow (2018)

[1] Kandala et al. (2017), Yeter-Aydeniz et al. (2019), LF et al. (2020)

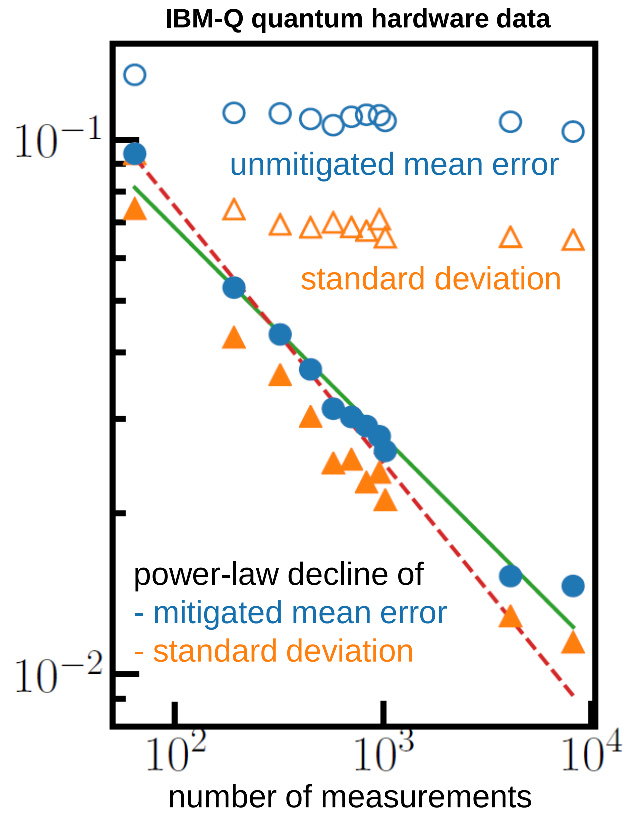
Error mitigation: how can we reduce the noise?

Example: measurement error mitigation	Example: gate error mitigation

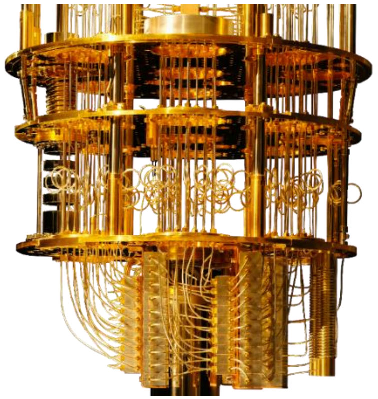
Operator rescaling method[1]

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

Result: measurement error reduced by factor 10



LF et al. (2020)



Carlow (2018)

Other mitigation techniques

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

[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),

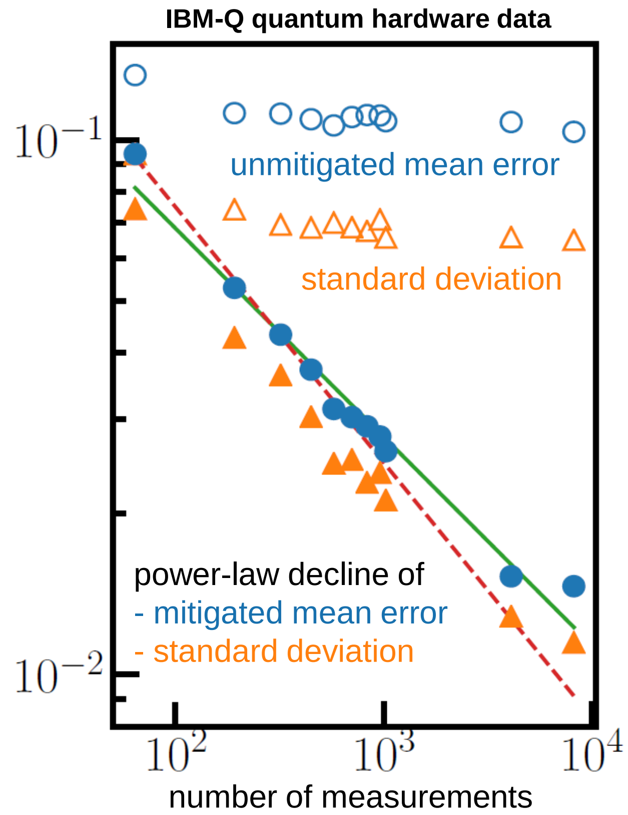
Error mitigation: how can we reduce the noise?

Example: measurement error mitigation

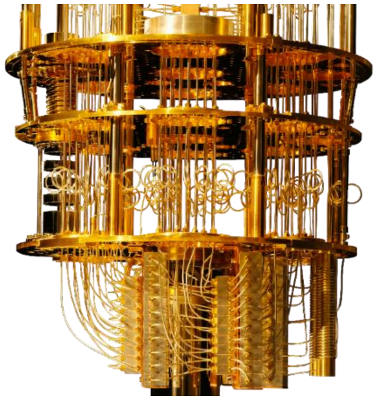
Operator rescaling method[1]

Benchmark: Z and Z_1Z_2 operators on IBM-Q hardware

Result: measurement error reduced by factor 10



LF et al. (2020)



Carlow (2018)

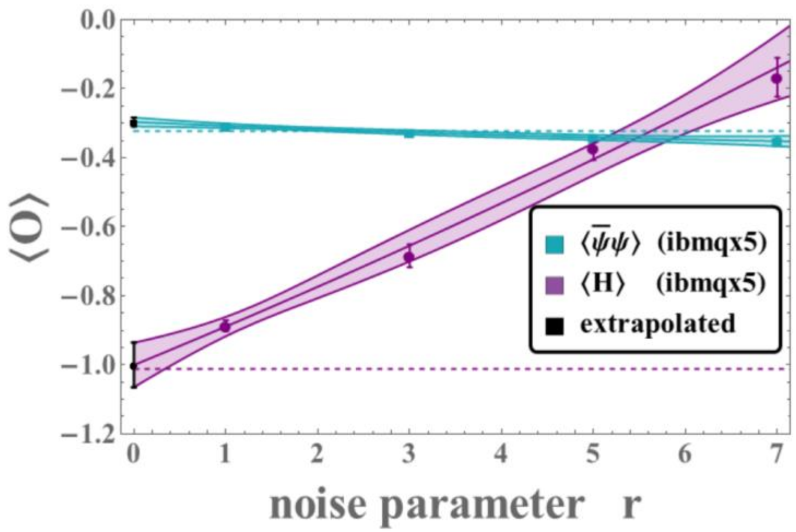
Example: gate error mitigation

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

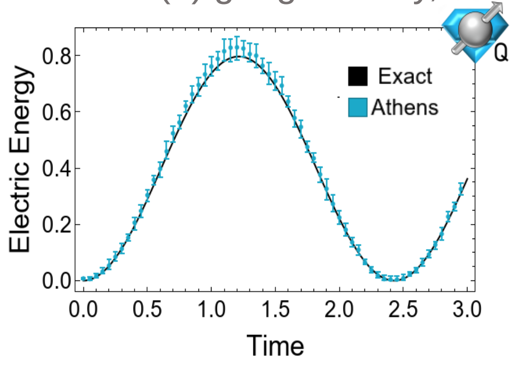
Which field theories have already been simulated?

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]



[1] Klco et al. (2018), de Jong et al. (2021), [2] Klco et al. (2019), [3] Ciavarella et al. (2019a,b)

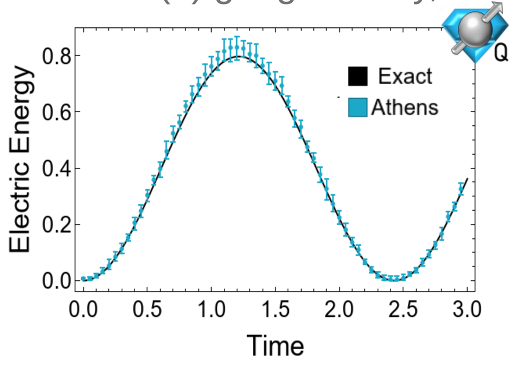
Which field theories have already been simulated?

Experimental results on "public" QC	Experimental results on "private" QC

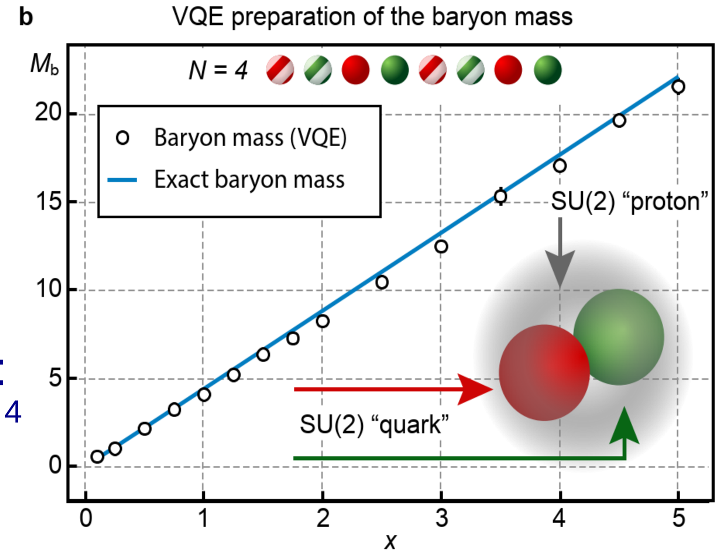
IBM-Q's superconducting qubits

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 [4]



[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)

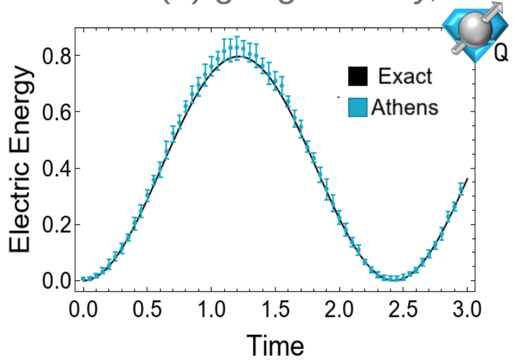
Which field theories have already been simulated?

Experimental results on "public" QC

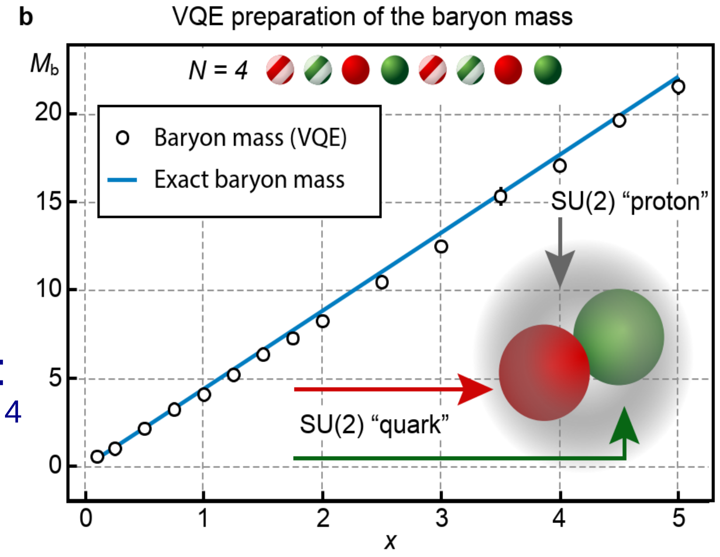
IBM-Q's superconducting qubits

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

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



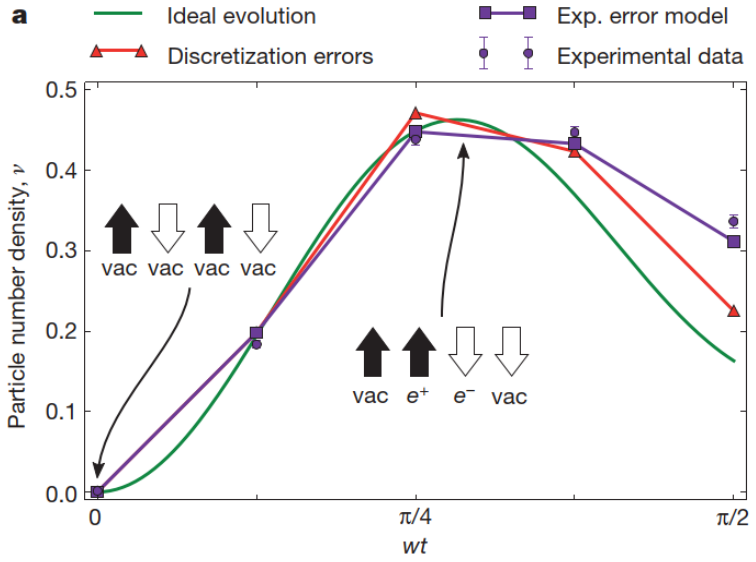
Variational computation: SU(2) "hadron" masses [4]



Experimental results on "private" QC

Trapped ions

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



Martinez et al. (2016)

[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), ...

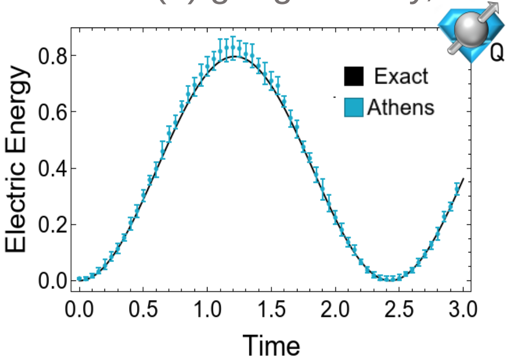
Which field theories have already been simulated?

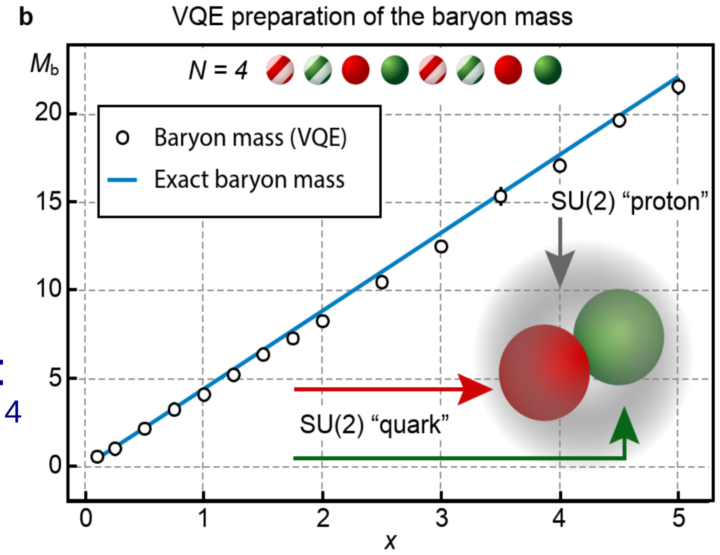
Experimental results on "public" QC	Experimental results on "private" QC

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]

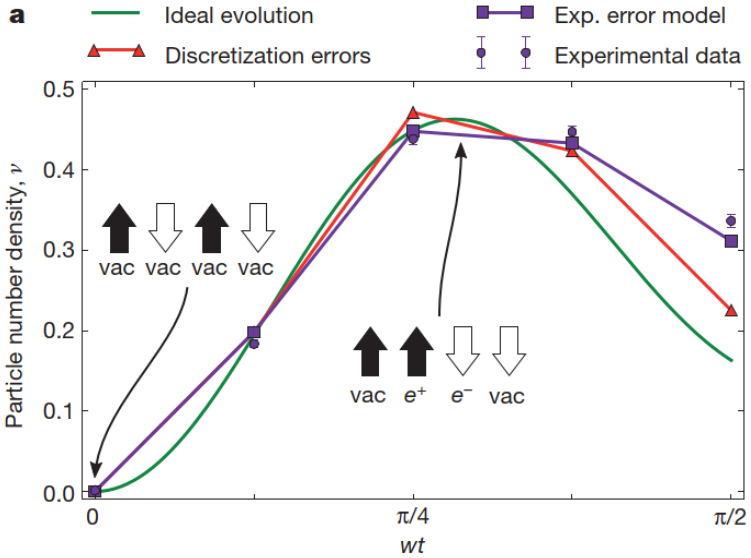




Variational computation:
SU(2) "hadron" masses [4]

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] Trotzky et al. (2012), [7] Yang et al. (2020), Mil et al. (2020), ...

How to simulate these field theories?

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

[1] Peruzzo et al. (2014)

How to simulate these field theories?

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

Key concept

Classical computer: main computation

[1] Peruzzo et al. (2014)

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

[1] Peruzzo et al. (2014)

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: even for small quantum hardware!

[1] Peruzzo et al. (2014)

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: even for small quantum hardware!

Goal

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

[1] Peruzzo et al. (2014)

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: even for small quantum hardware!

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 $\vec{\alpha}$

[1] Peruzzo et al. (2014)

How to simulate these field theories?

Example: hybrid quantum-classical algorithms

Key concept

Classical computer: main computation

Quantum computer: classically hard/intractable part

Advantages: even for small quantum hardware!

Variational Quantum Eigensolver (VQE) [1]

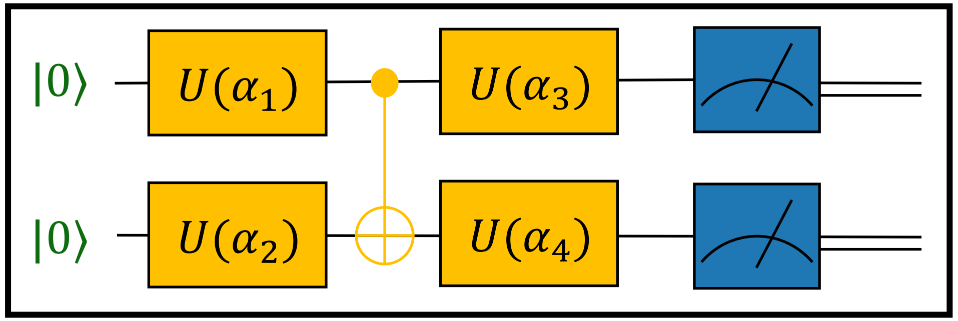
Goal

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 $\vec{\alpha}$

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



[1] Peruzzo et al. (2014)

How to simulate these field theories?

Example: hybrid quantum-classical algorithms

Key concept

Classical computer: main computation

Quantum computer: classically hard/intractable part

Advantages: even for small quantum hardware!

Variational Quantum Eigensolver (VQE) [1]

Goal

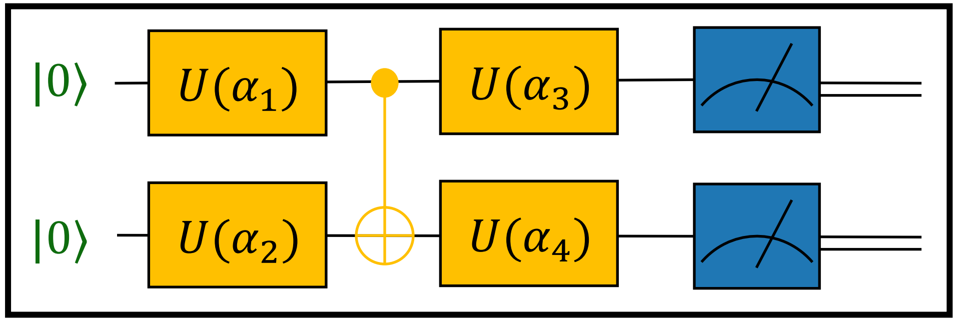
Find ground state and excited states of Hamiltonian $\mathcal{H}$

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 $\vec{\alpha}_{i+1}$

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

[1] Peruzzo et al. (2014)

How to simulate these field theories?

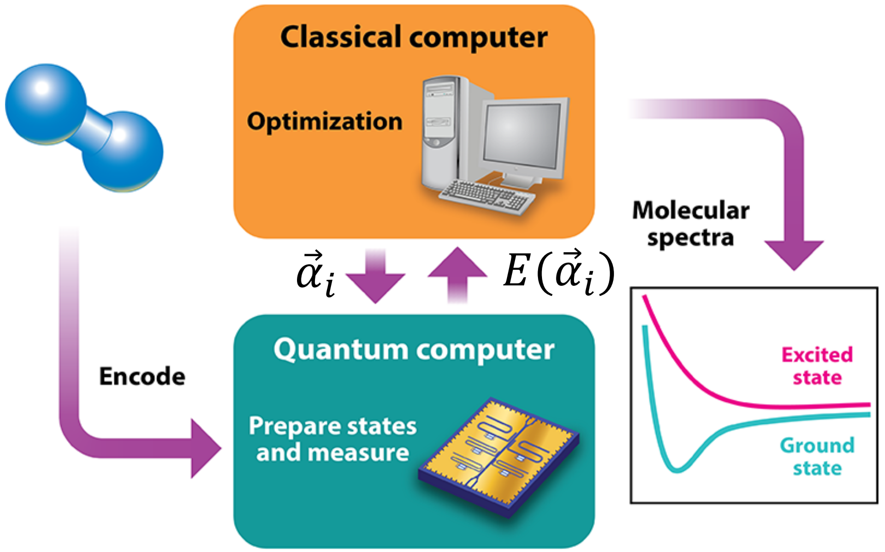
Example: hybrid quantum-classical algorithms	Variational Quantum Eigensolver (VQE) [1]

Key concept

Classical computer: main computation

Quantum computer: classically hard/intractable part

Advantages: even for small quantum hardware!



Sim et al. (2018)

Goal

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

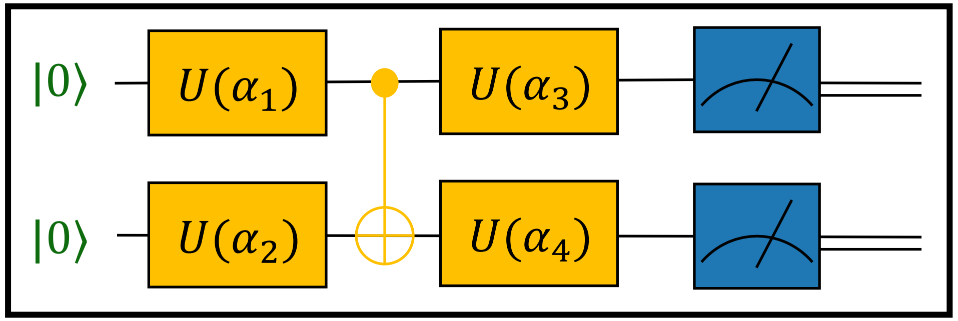
Variational approach

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 $\vec{\alpha}_{i+1}$

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



[1] Peruzzo et al. (2014)

How to simulate these field theories?

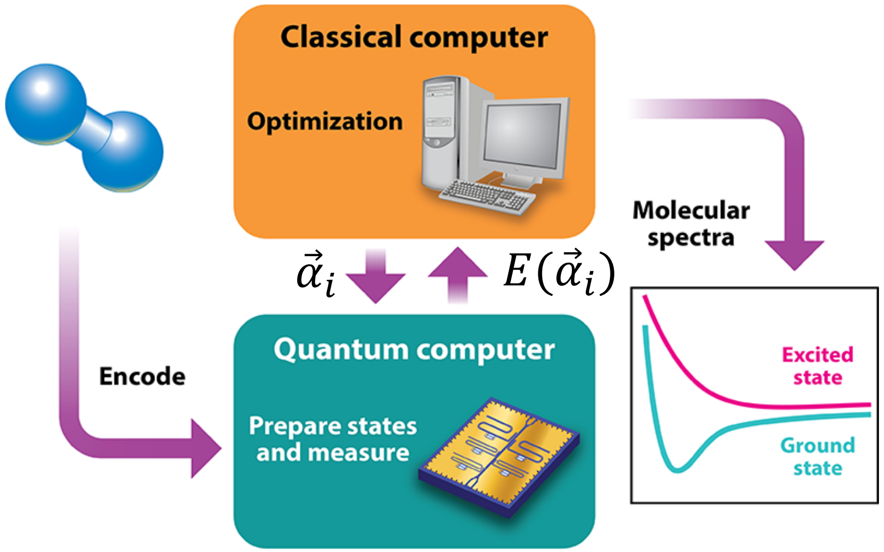
Example: hybrid quantum-classical algorithms

Key concept

Classical computer: main computation

Quantum computer: classically hard/intractable part

Advantages: even for small quantum hardware!



Sim et al. (2018)

Variational Quantum Eigensolver (VQE) [1]

Goal

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

Variational approach

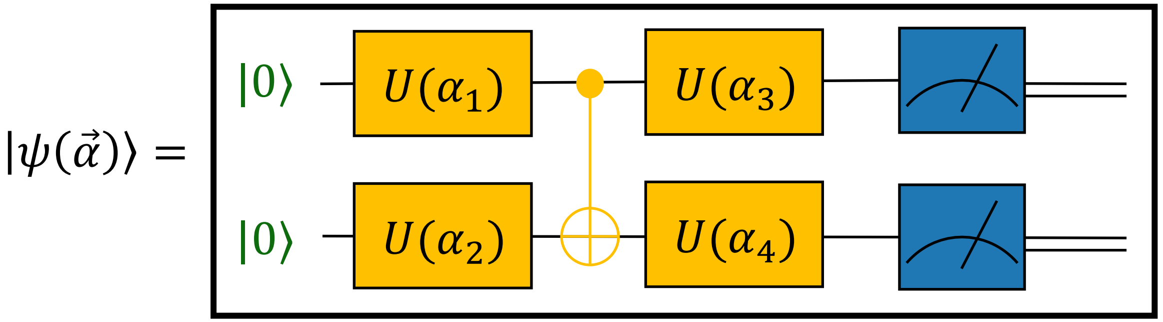
Minimize $E(\vec{\alpha}) = \langle\psi(\vec{\alpha})|\mathcal{H}|\psi(\vec{\alpha})\rangle$ w.r.t. parameters $\vec{\alpha}$

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 $E(\vec{\alpha}_i)$



[1] Peruzzo et al. (2014)

How to simulate these field theories?

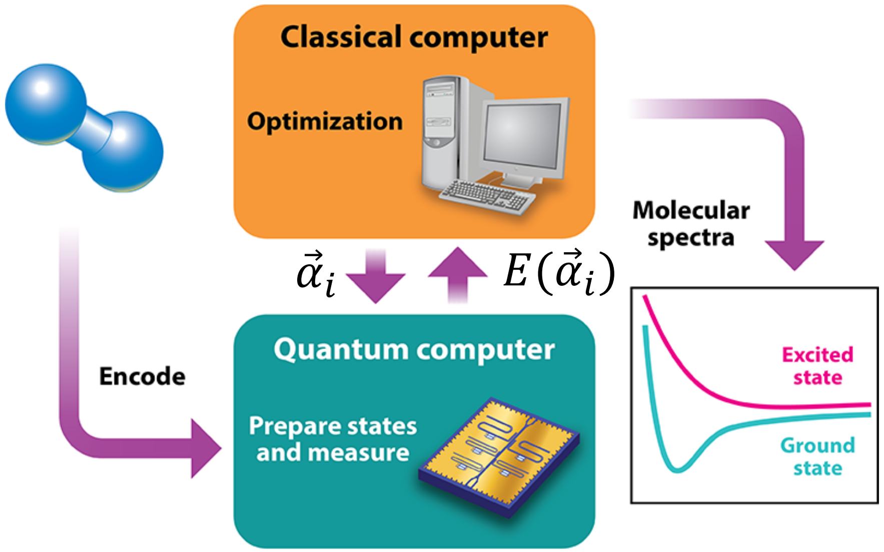
Example: hybrid quantum-classical algorithms

Key concept

Classical computer: main computation

Quantum computer: classically hard/intractable part

Advantages: even for small quantum hardware!



Sim et al. (2018)

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 $\vec{\alpha}$

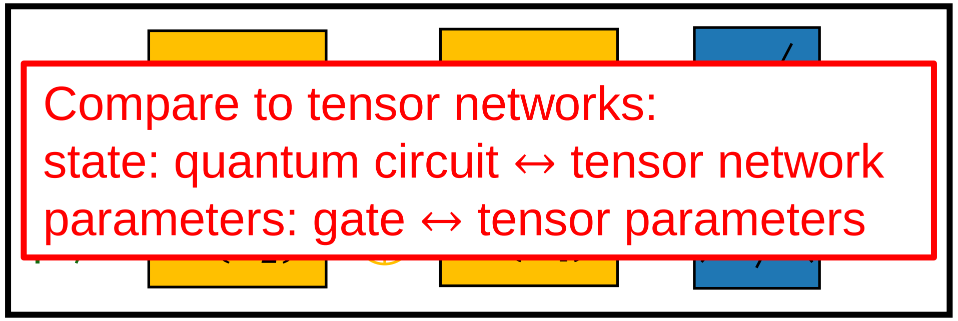
Classical computer

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 $E(\vec{\alpha}_i)$

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



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]	**Example: geometrical method** [2]

[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)

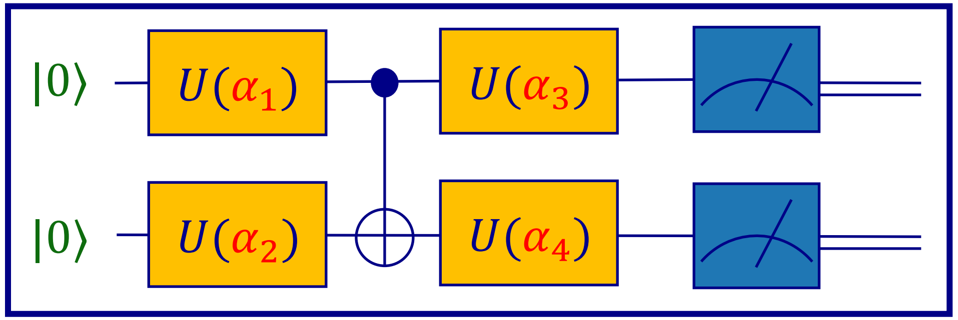
How to prepare the quantum state?

Quantum circuit design [1] | **Example: geometrical method** [2]

Maximal expressivity

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

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



[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 prepare the quantum state?

Quantum circuit design [1] | **Example: geometrical method** [2]

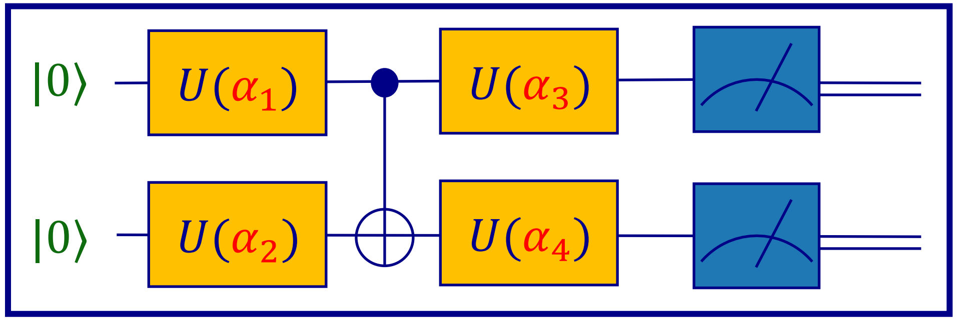
Maximal expressivity

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

Minimality

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

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



[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 prepare the quantum state?

Quantum circuit design [1]

Example: geometrical method [2]

Maximal expressivity

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

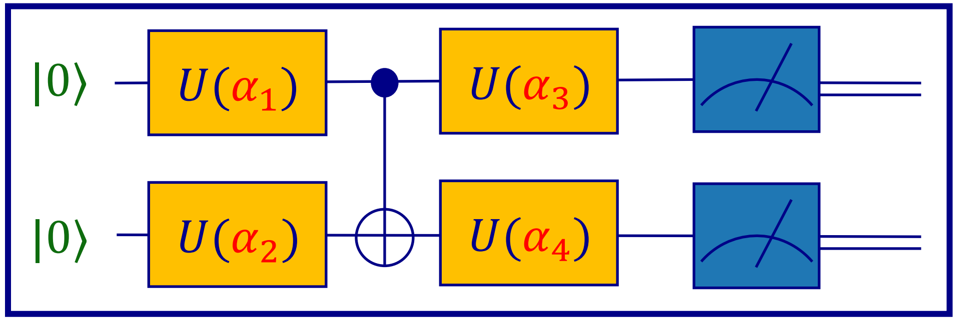
Minimality

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

Symmetry

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

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



[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 prepare the quantum state?

Quantum circuit design [1]

Maximal expressivity

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

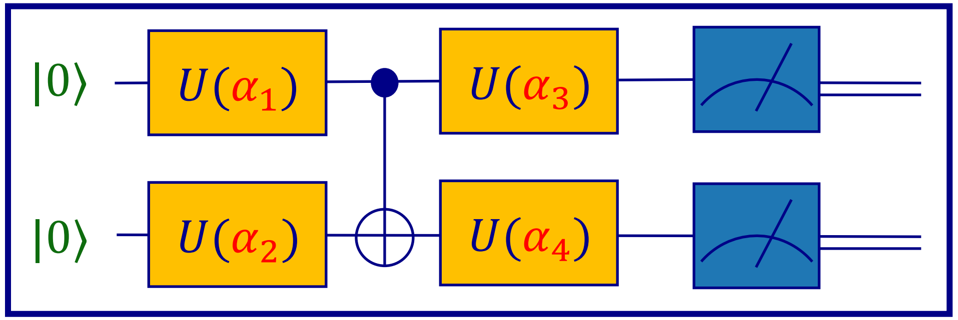
Minimality

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

[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 prepare the quantum state?

Quantum circuit design [1]

Example: geometrical method [2]

Maximal expressivity

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

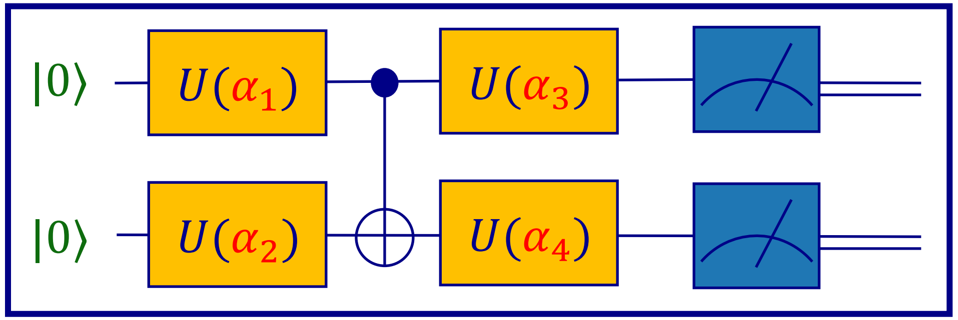
Minimality

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

Symmetry

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

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



Manifolds

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

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

[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 prepare the quantum state?

Quantum circuit design [1]

Maximal expressivity

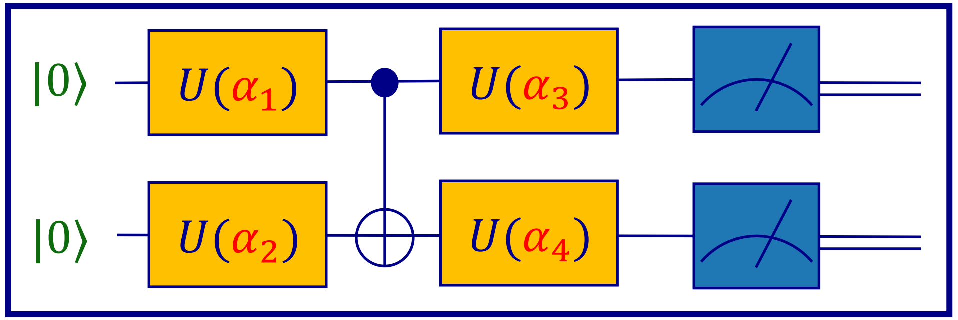
$|\psi(\vec{\alpha})\rangle$ should reach all physical states in Hilbert space

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

[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 prepare the quantum state?

Quantum circuit design [1]

Maximal expressivity

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

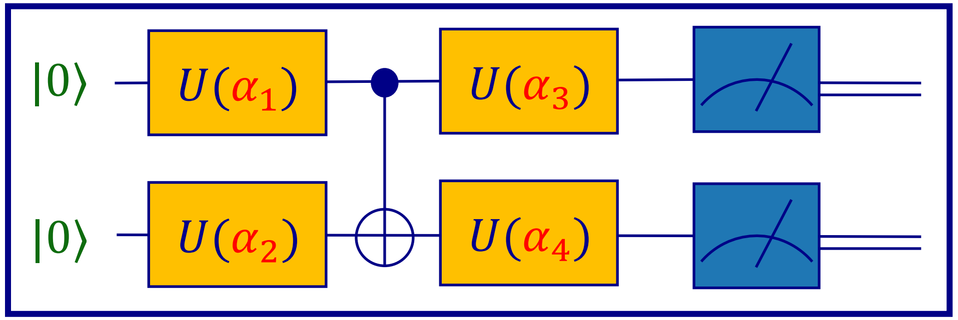
Minimality

$|\psi(\vec{\alpha})\rangle$ 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: $\mathrm{codim}(M) = \dim(S) - \dim(M) \overset{!}{=} 0$

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

[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 prepare the quantum state?

Quantum circuit design [1]

Maximal expressivity

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

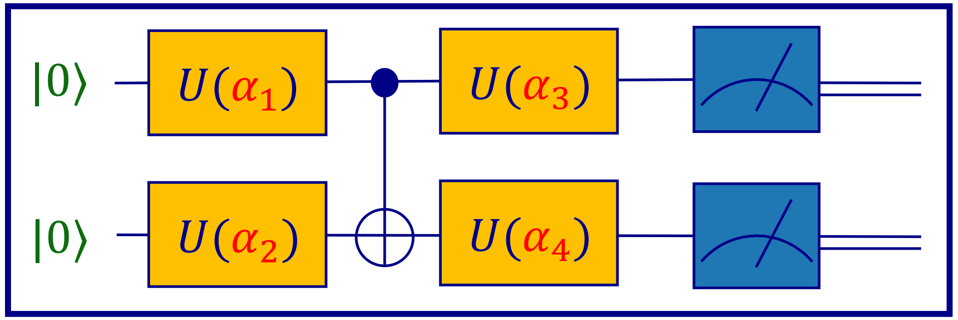
Minimality

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

Symmetry

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

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



Example: geometrical method [2]

Manifolds

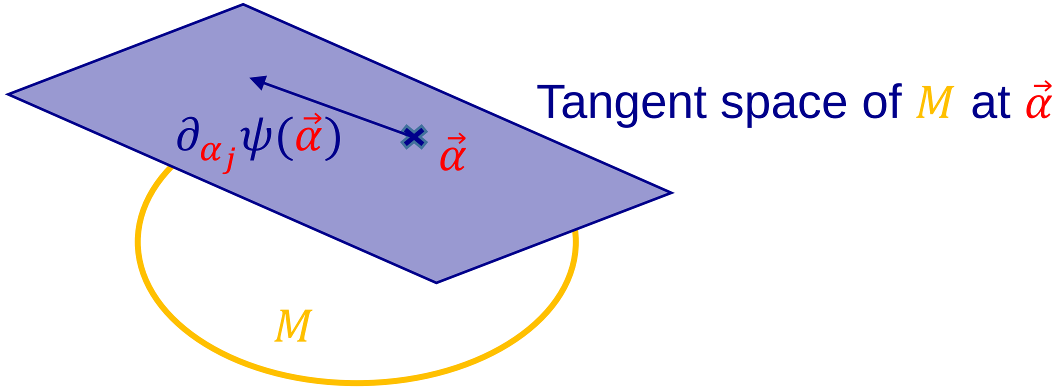
Circuit manifold M: states $|\psi(\vec{\alpha})\rangle$ reachable by circuit

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

Optimization

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

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



[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 prepare the quantum state?

Quantum circuit design [1]

Maximal expressivity

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

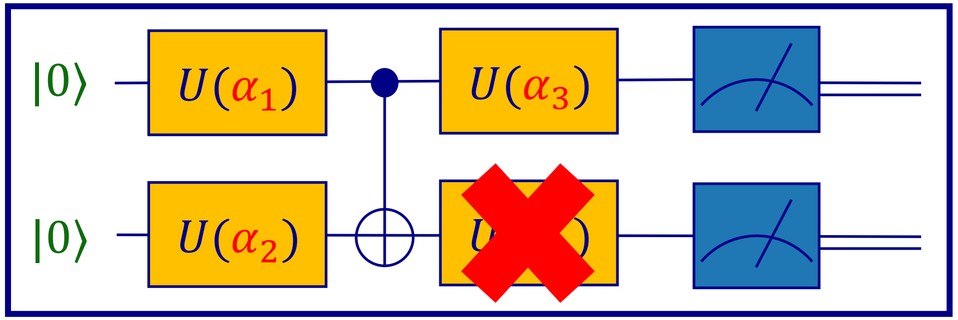
Minimality

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

Symmetry

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

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



Example: geometrical method [2]

Manifolds

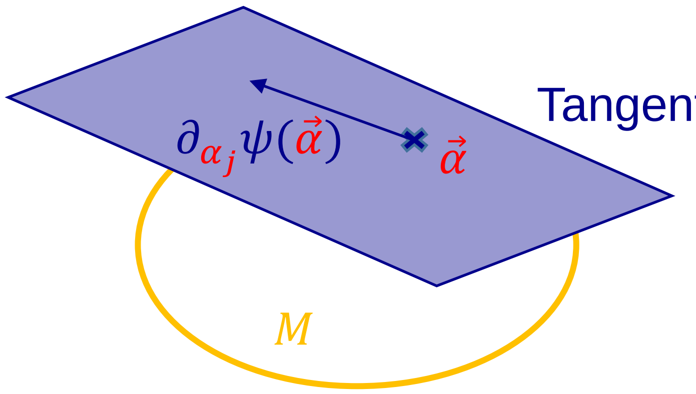
Circuit manifold M: states $|\psi(\vec{\alpha})\rangle$ reachable by circuit

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 deal with gauge fields?

Infinite Hilbert space	Gauge invariance

How to deal with gauge fields?

Infinite Hilbert space	Gauge invariance

Problem

Continuous gauge theory requires ∞-dim. Hilbert space

How to deal with gauge fields?

Infinite Hilbert space	Gauge invariance

Problem

Continuous gauge theory requires ∞-dim. Hilbert space

First approach

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

How to deal with gauge fields?

Infinite Hilbert space	Gauge invariance

Problem

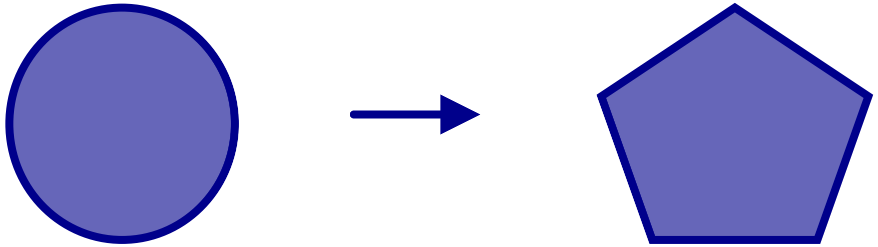
Continuous gauge theory requires ∞-dim. Hilbert space

First approach

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

Second approach

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



[1] Zohar et al. (2013), ...

How to deal with gauge fields?

Infinite Hilbert space | **Gauge invariance**

Problem

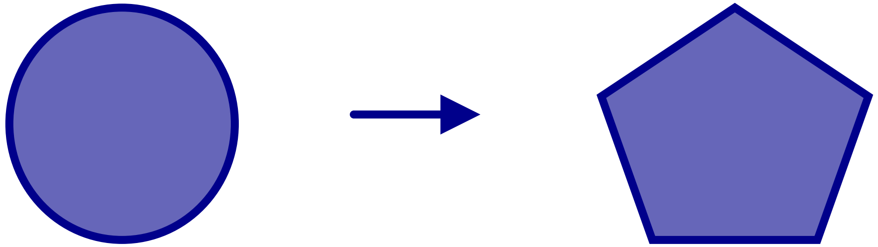
Continuous gauge theory requires ∞-dim. Hilbert space

First approach

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

Second approach

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



Third approach

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

[1] Zohar et al. (2013), ..., [2] Horn (1981), ...

How to deal with gauge fields?

Infinite Hilbert space	Gauge invariance

Problem

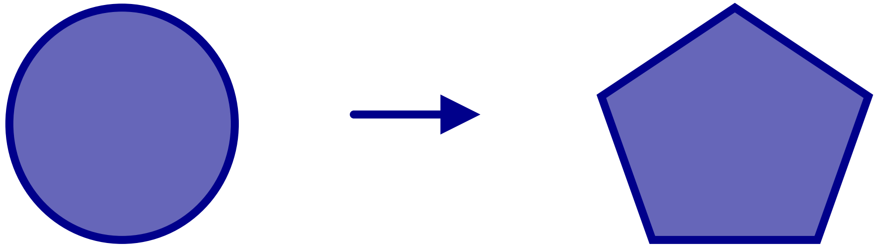
Continuous gauge theory requires ∞-dim. Hilbert space

First approach

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

Second approach

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



Third approach

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

Many more approaches → see parallel talks!

[1] Zohar et al. (2013), ..., [2] Horn (1981), ...

How to deal with gauge fields?

Infinite Hilbert space

Problem

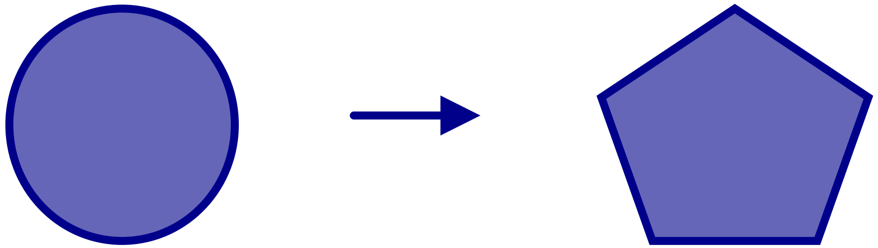
Continuous gauge theory requires ∞-dim. Hilbert space

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 → see parallel talks!

Gauge invariance

Problem

Gauge invariance requires imposing local constraints

[1] Zohar et al. (2013), ..., [2] Horn (1981), ...

How to deal with gauge fields?

Infinite Hilbert space

Problem

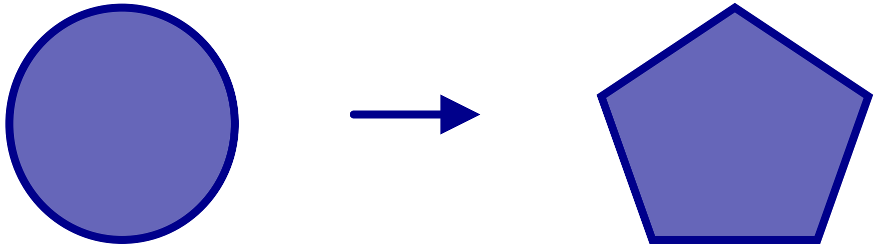
Continuous gauge theory requires ∞-dim. Hilbert space

First approach

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

Second approach

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



Third approach

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

Many more approaches → see parallel talks!

Gauge invariance

Problem

Gauge invariance requires imposing local constraints

First approach

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

[1] Zohar et al. (2013), ..., [2] Horn (1981), ..., [3] Banerjee et al. (2012), ...

How to deal with gauge fields?

Infinite Hilbert space

Problem

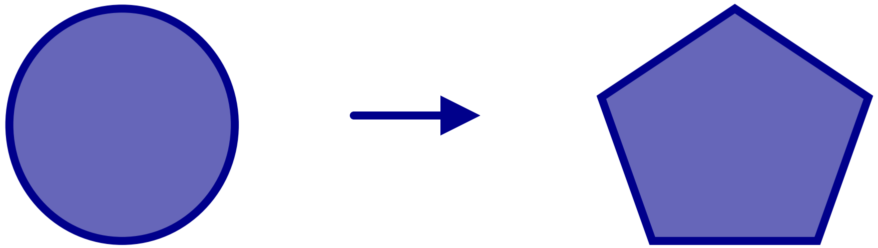
Continuous gauge theory requires ∞-dim. Hilbert space

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 → see parallel talks!

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]

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

How to deal with gauge fields?

Infinite Hilbert space

Problem

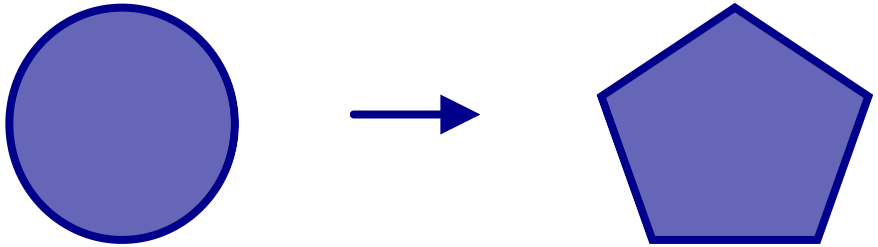
Continuous gauge theory requires ∞-dim. Hilbert space

First approach

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

Second approach

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



Third approach

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

Many more approaches → see parallel talks!

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]

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

How to deal with gauge fields?

Infinite Hilbert space

Problem

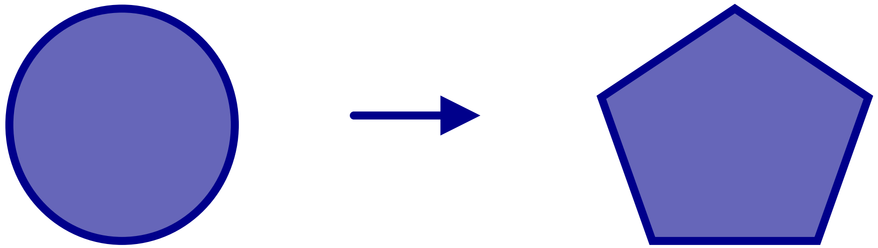
Continuous gauge theory requires ∞-dim. Hilbert space

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 → see parallel talks!

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 → see parallel talks!

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

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D	Discrete and higher-dimensional gauge theories

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D	Discrete and higher-dimensional gauge theories

1+1D Schwinger model with chemical potential

Studying multi-flavor chemical potential with VQE

→ talk by Stefan Kühn tomorrow at 2:20pm

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D	Discrete and higher-dimensional gauge theories

1+1D Schwinger model with chemical potential

Studying multi-flavor chemical potential with VQE

→ talk by Stefan Kühn tomorrow at 2:20pm

1+1D Schwinger model with θ-term

Comparing quantum algorithms for state preparation

→ talk by Alexei Bazavov tomorrow at 3:20pm

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D	Discrete and higher-dimensional gauge theories

1+1D Schwinger model with chemical potential

Studying multi-flavor chemical potential with VQE

→ talk by Stefan Kühn tomorrow at 2:20pm

1+1D Schwinger model with θ-term

Comparing quantum algorithms for state preparation

→ talk by Alexei Bazavov tomorrow at 3:20pm

1+1D Schwinger model with Wilson fermions

Comparing Wilson and staggered fermions

→ talk by Takis Angelides tomorrow at 5:30pm

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D	Discrete and higher-dimensional gauge theories

1+1D Schwinger model with chemical potential

Studying multi-flavor chemical potential with VQE

→ talk by Stefan Kühn tomorrow at 2:20pm

1+1D Schwinger model with θ-term

Comparing quantum algorithms for state preparation

→ talk by Alexei Bazavov tomorrow at 3:20pm

1+1D Schwinger model with Wilson fermions

Comparing Wilson and staggered fermions

→ talk by Takis Angelides tomorrow at 5:30pm

1+1D Schwinger model at finite temperature

Mapping out $T - \mu$ phase diagram with VQE

→ talk by Akio Tomiya on Thursday at 12:30pm

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D

1+1D Schwinger model with chemical potential

Studying multi-flavor chemical potential with VQE

→ talk by Stefan Kühn tomorrow at 2:20pm

1+1D Schwinger model with θ-term

Comparing quantum algorithms for state preparation

→ talk by Alexei Bazavov tomorrow at 3:20pm

1+1D Schwinger model with Wilson fermions

Comparing Wilson and staggered fermions

→ talk by Takis Angelides tomorrow at 5:30pm

1+1D Schwinger model at finite temperature

Mapping out $T - \mu$ phase diagram with VQE

→ talk by Akio Tomiya on Thursday at 12:30pm

Discrete and higher-dimensional gauge theories

1+1D D_n gauge theory

Preparing ground state & evolution on quantum annealer

→ talk by Michael Fromm on Wednesday at 2pm

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D

1+1D Schwinger model with chemical potential

Studying multi-flavor chemical potential with VQE

→ talk by Stefan Kühn tomorrow at 2:20pm

1+1D Schwinger model with θ-term

Comparing quantum algorithms for state preparation

→ talk by Alexei Bazavov tomorrow at 3:20pm

1+1D Schwinger model with Wilson fermions

Comparing Wilson and staggered fermions

→ talk by Takis Angelides tomorrow at 5:30pm

1+1D Schwinger model at finite temperature

Mapping out $T - \mu$ phase diagram with VQE

→ talk by Akio Tomiya on Thursday at 12:30pm

Discrete and higher-dimensional gauge theories

1+1D D_n gauge theory

Preparing ground state & evolution on quantum annealer

→ talk by Michael Fromm on Wednesday at 2pm

1+1D Z_2 gauge theory

Developing quantum algorithms for thermal states

→ talk by Connor Powers tomorrow at 3:20pm

More details: parallel talks @ this conference

Quantum Electrodynamics in 1+1D	Discrete and higher-dimensional gauge theories
1+1D Schwinger model with chemical potential	**1+1D D_n gauge theory**
Studying multi-flavor chemical potential with VQE	Preparing ground state & evolution on quantum annealer
→ talk by Stefan Kühn tomorrow at 2:20pm	→ talk by Michael Fromm on Wednesday at 2pm
1+1D Schwinger model with θ-term	**1+1D Z_2 gauge theory**
Comparing quantum algorithms for state preparation	Developing quantum algorithms for thermal states
→ talk by Alexei Bazavov tomorrow at 3:20pm	→ talk by Connor Powers tomorrow at 3:20pm
1+1D Schwinger model with Wilson fermions	**2+1D U(1) gauge theory**
Comparing Wilson and staggered fermions	Finding a resource-efficient implementation
→ talk by Takis Angelides tomorrow at 5:30pm	→ talk by Christopher Kane tomorrow at 2pm
1+1D Schwinger model at finite temperature	
Mapping out $T - \mu$ phase diagram with VQE	
→ talk by Akio Tomiya on Thursday at 12:30pm	

More details: parallel talks @ this conference

SU(N) lattice gauge theories	Lattice field theories

More details: parallel talks @ this conference

SU(N) lattice gauge theories	Lattice field theories
1+1D SU(2) gauge theory Implementing real-time evolution on IBM-Q hardware → talk by Emanuele Mendicelli tomorrow at 4:30pm	

More details: parallel talks @ this conference

SU(N) lattice gauge theories	Lattice field theories

1+1D SU(2) gauge theory

Implementing real-time evolution on IBM-Q hardware

→ talk by Emanuele Mendicelli tomorrow at 4:30pm

Comparing Schwinger-boson & loop-string hadron form.

→ talk by Jesse Stryker on Thursday at 11:50am

More details: parallel talks @ this conference

SU(N) lattice gauge theories	Lattice field theories

1+1D SU(2) gauge theory

Implementing real-time evolution on IBM-Q hardware

→ talk by Emanuele Mendicelli tomorrow at 4:30pm

Comparing Schwinger-boson & loop-string hadron form.

→ talk by Jesse Stryker on Thursday at 11:50am

SU(2) gauge theories

Developing efficient digitization via finite subgroups

→ talk by Timo Jakobs tomorrow at 2:40pm

More details: parallel talks @ this conference

SU(N) lattice gauge theories	Lattice field theories

1+1D SU(2) gauge theory

Implementing real-time evolution on IBM-Q hardware

→ talk by Emanuele Mendicelli tomorrow at 4:30pm

Comparing Schwinger-boson & loop-string hadron form.

→ talk by Jesse Stryker on Thursday at 11:50am

SU(2) gauge theories

Developing efficient digitization via finite subgroups

→ talk by Timo Jakobs tomorrow at 2:40pm

Defining canonical momenta for discretized gauge fields

→ talk by Carsten Urbach tomorrow at 3pm

More details: parallel talks @ this conference

SU(N) lattice gauge theories	**Lattice field theories**

1+1D SU(2) gauge theory

Implementing real-time evolution on IBM-Q hardware

→ talk by Emanuele Mendicelli tomorrow at 4:30pm

Comparing Schwinger-boson & loop-string hadron form.

→ talk by Jesse Stryker on Thursday at 11:50am

SU(2) gauge theories

Developing efficient digitization via finite subgroups

→ talk by Timo Jakobs tomorrow at 2:40pm

Defining canonical momenta for discretized gauge fields

→ talk by Carsten Urbach tomorrow at 3pm

SU(3) gauge theories

Developing efficient digitization & improved Hamiltonians

→ talk by Henry Lamm tomorrow at 5:50pm

More details: parallel talks @ this conference

SU(N) lattice gauge theories

1+1D SU(2) gauge theory

Implementing real-time evolution on IBM-Q hardware

→ talk by Emanuele Mendicelli tomorrow at 4:30pm

Comparing Schwinger-boson & loop-string hadron form.

→ talk by Jesse Stryker on Thursday at 11:50am

SU(2) gauge theories

Developing efficient digitization via finite subgroups

→ talk by Timo Jakobs tomorrow at 2:40pm

Defining canonical momenta for discretized gauge fields

→ talk by Carsten Urbach tomorrow at 3pm

SU(3) gauge theories

Developing efficient digitization & improved Hamiltonians

→ talk by Henry Lamm tomorrow at 5:50pm

Lattice field theories

1+1D Ising model

Studying NISQ algorithms for open quantum systems

→ talk by Bharath Sambasivam tomorrow at 6:10pm

More details: parallel talks @ this conference

SU(N) lattice gauge theories	Lattice field theories

1+1D SU(2) gauge theory

Implementing real-time evolution on IBM-Q hardware

→ talk by Emanuele Mendicelli tomorrow at 4:30pm

Comparing Schwinger-boson & loop-string hadron form.

→ talk by Jesse Stryker on Thursday at 11:50am

SU(2) gauge theories

Developing efficient digitization via finite subgroups

→ talk by Timo Jakobs tomorrow at 2:40pm

Defining canonical momenta for discretized gauge fields

→ talk by Carsten Urbach tomorrow at 3pm

SU(3) gauge theories

Developing efficient digitization & improved Hamiltonians

→ talk by Henry Lamm tomorrow at 5:50pm

1+1D Ising model

Studying NISQ algorithms for open quantum systems

→ talk by Bharath Sambasivam tomorrow at 6:10pm

1+1D extended O(2) model

Studying discrete approximations of continuous groups

→ talk by Leon Hostetler on Wednesday at 3:40pm

More details: parallel talks @ this conference

SU(N) lattice gauge theories

1+1D SU(2) gauge theory

Implementing real-time evolution on IBM-Q hardware

→ talk by Emanuele Mendicelli tomorrow at 4:30pm

Comparing Schwinger-boson & loop-string hadron form.

→ talk by Jesse Stryker on Thursday at 11:50am

SU(2) gauge theories

Developing efficient digitization via finite subgroups

→ talk by Timo Jakobs tomorrow at 2:40pm

Defining canonical momenta for discretized gauge fields

→ talk by Carsten Urbach tomorrow at 3pm

SU(3) gauge theories

Developing efficient digitization & improved Hamiltonians

→ talk by Henry Lamm tomorrow at 5:50pm

Lattice field theories

1+1D Ising model

Studying NISQ algorithms for open quantum systems

→ talk by Bharath Sambasivam tomorrow at 6:10pm

1+1D extended O(2) model

Studying discrete approximations of continuous groups

→ talk by Leon Hostetler on Wednesday at 3:40pm

1+1D Wess-Zumino model

Studying dynamical breaking of supersymmetry

→ talk by Christopher Culver tomorrow at 2:20pm

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

The Path to Go…	A Rough Sketch…

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

The Path to Go…	A Rough Sketch…

Present

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

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

The Path to Go…	A Rough Sketch…

Present

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

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

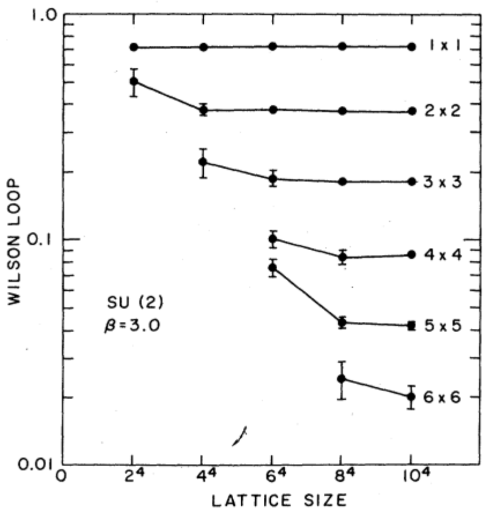
Summary: where do we stand, where will we go?

The Path to Go…	A Rough Sketch…

Present

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

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



Creutz (1980)

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

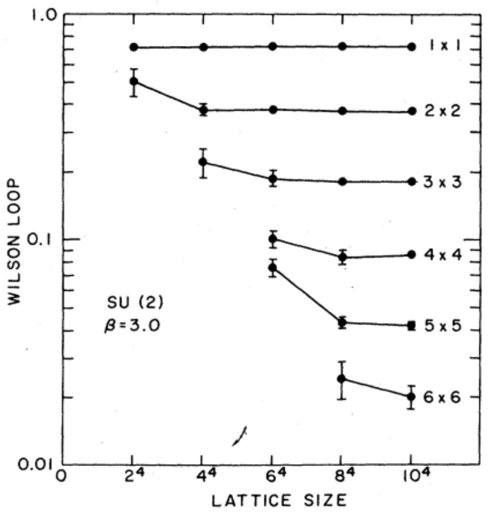
The Path to Go…

A Rough Sketch…

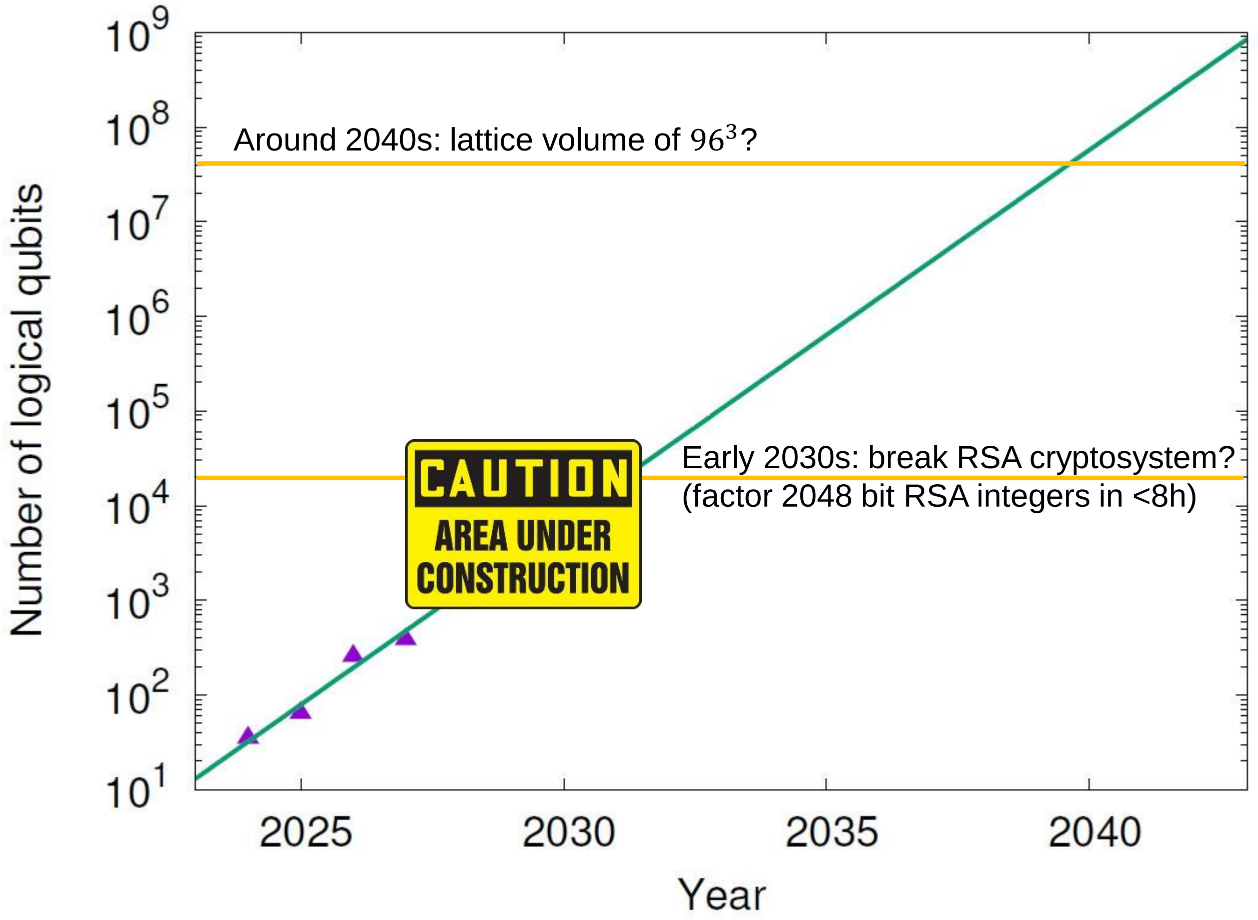
Present

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

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



Creutz (1980)



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

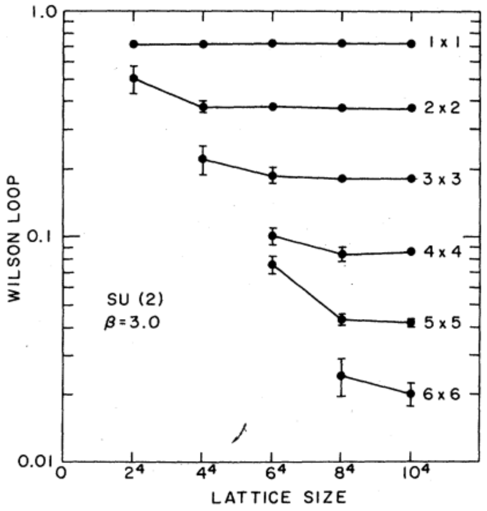
Summary: where do we stand, where will we go?

The Path to Go…

Present

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

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

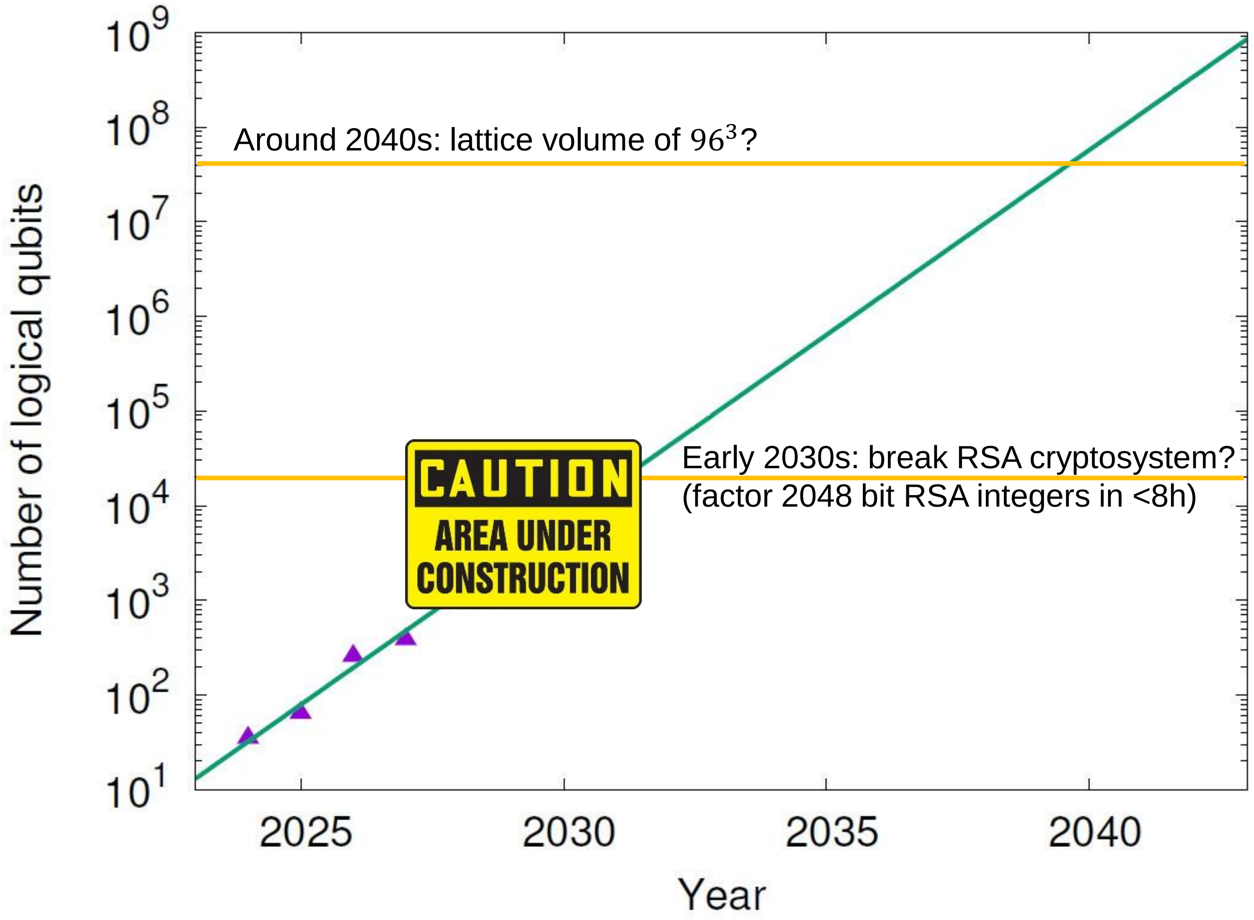


Creutz (1980)

Future

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

A Rough Sketch…



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

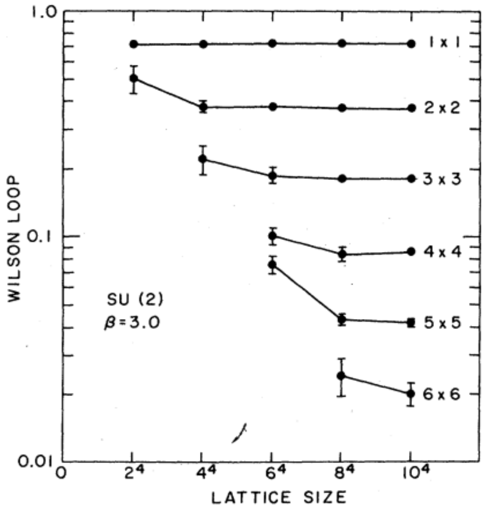
Summary: where do we stand, where will we go?

The Path to Go…	A Rough Sketch…

Present

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

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

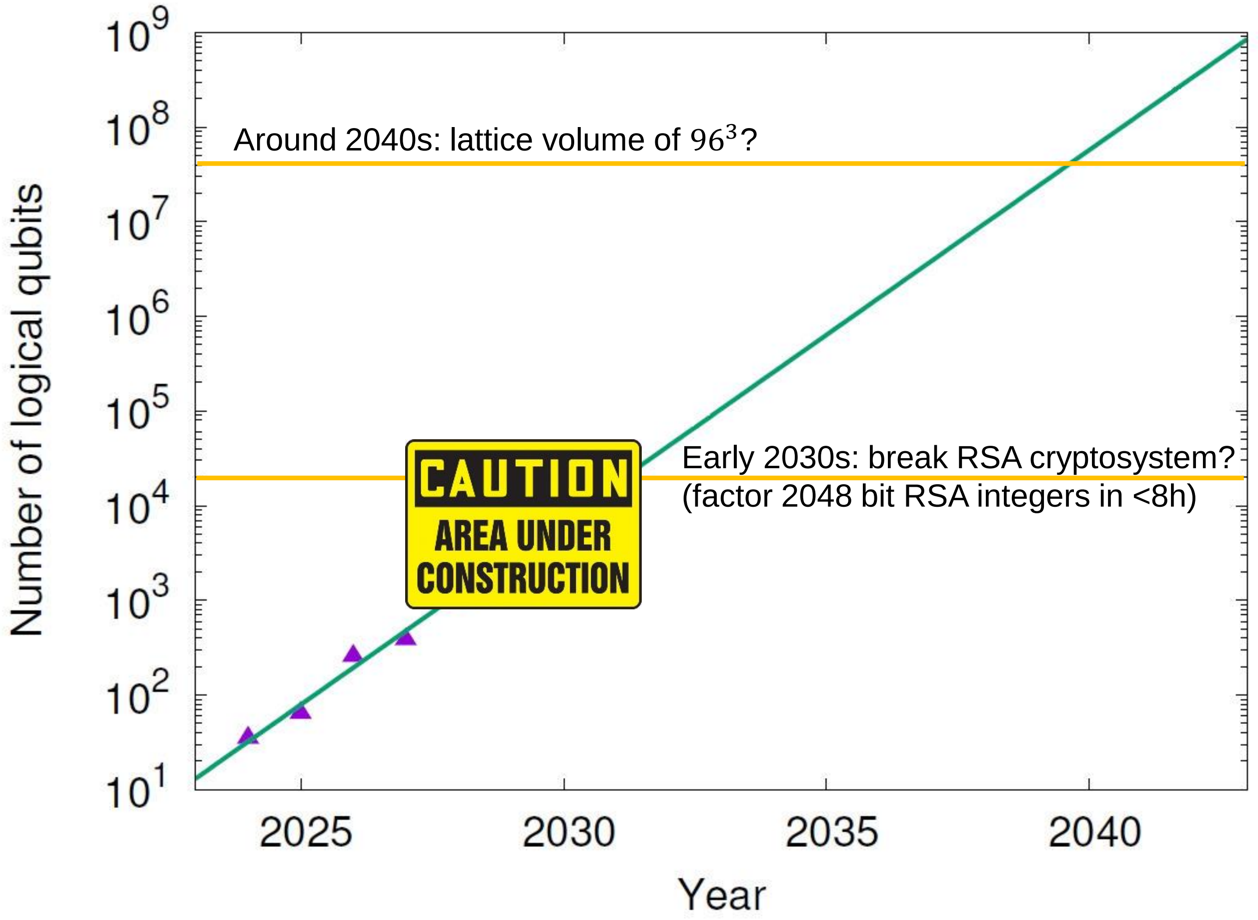


Creutz (1980)

Future

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

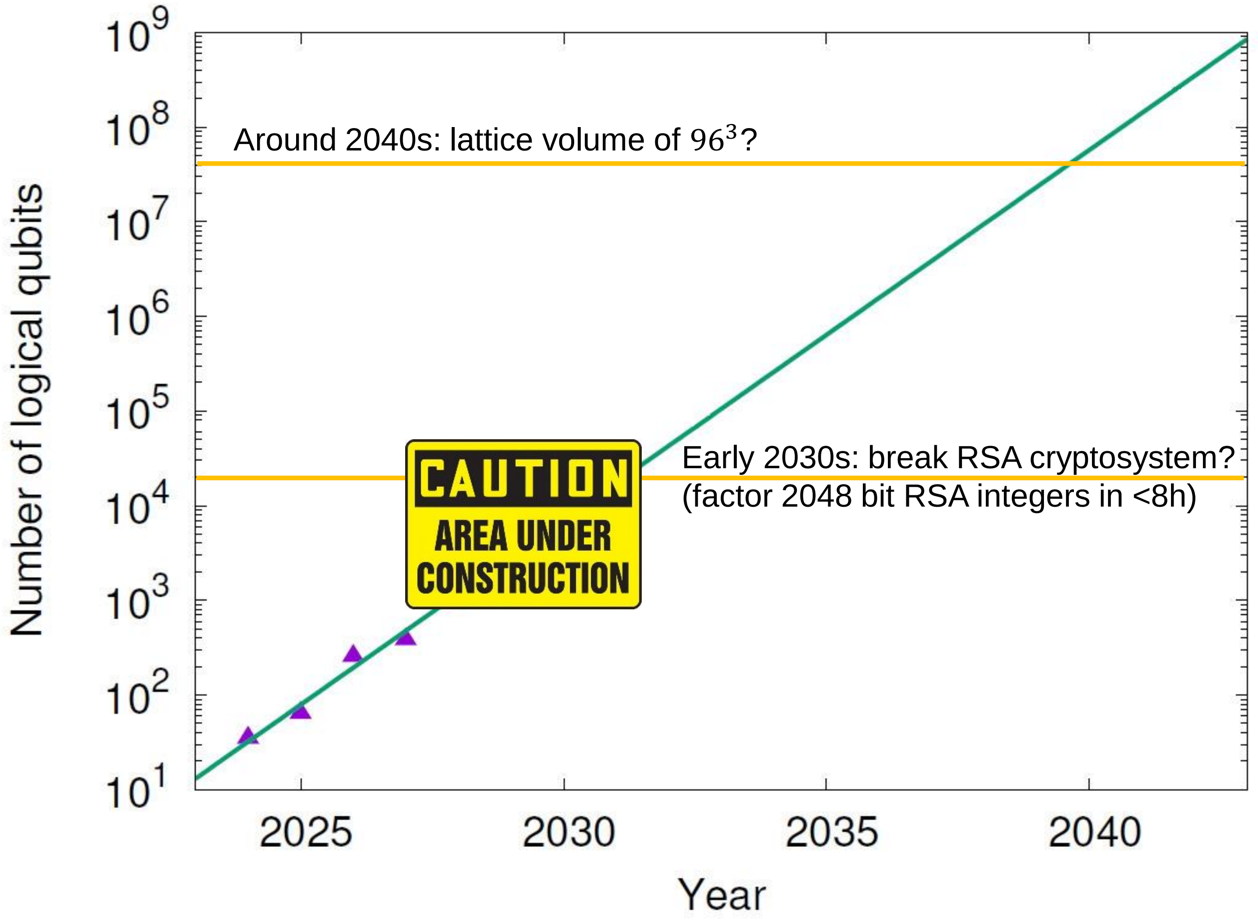
Algorithms: improved Hamiltonians, resource-efficient formulations of 3+1D LGTs, …



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

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

The Way Forward…	A Rough Sketch…



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

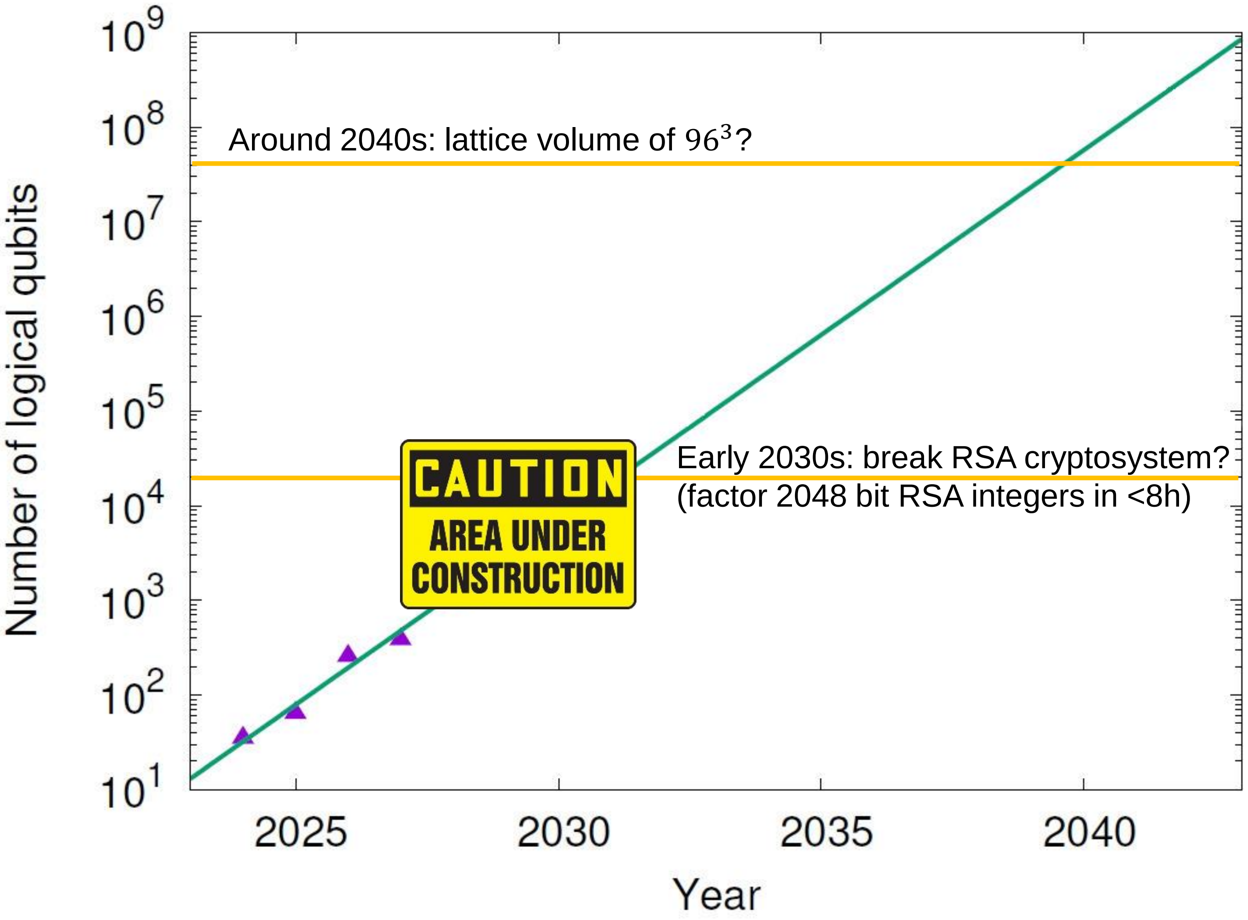
Summary: where do we stand, where will we go?

The Way Forward…	A Rough Sketch…

Quantum

new and quickly progressing

→ high risk, high gain



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

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

The Way Forward…	A Rough Sketch…

Quantum

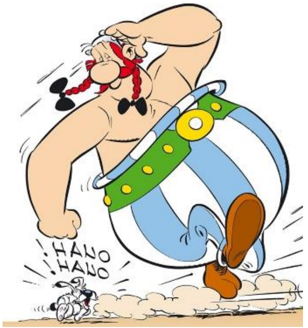
new and quickly progressing

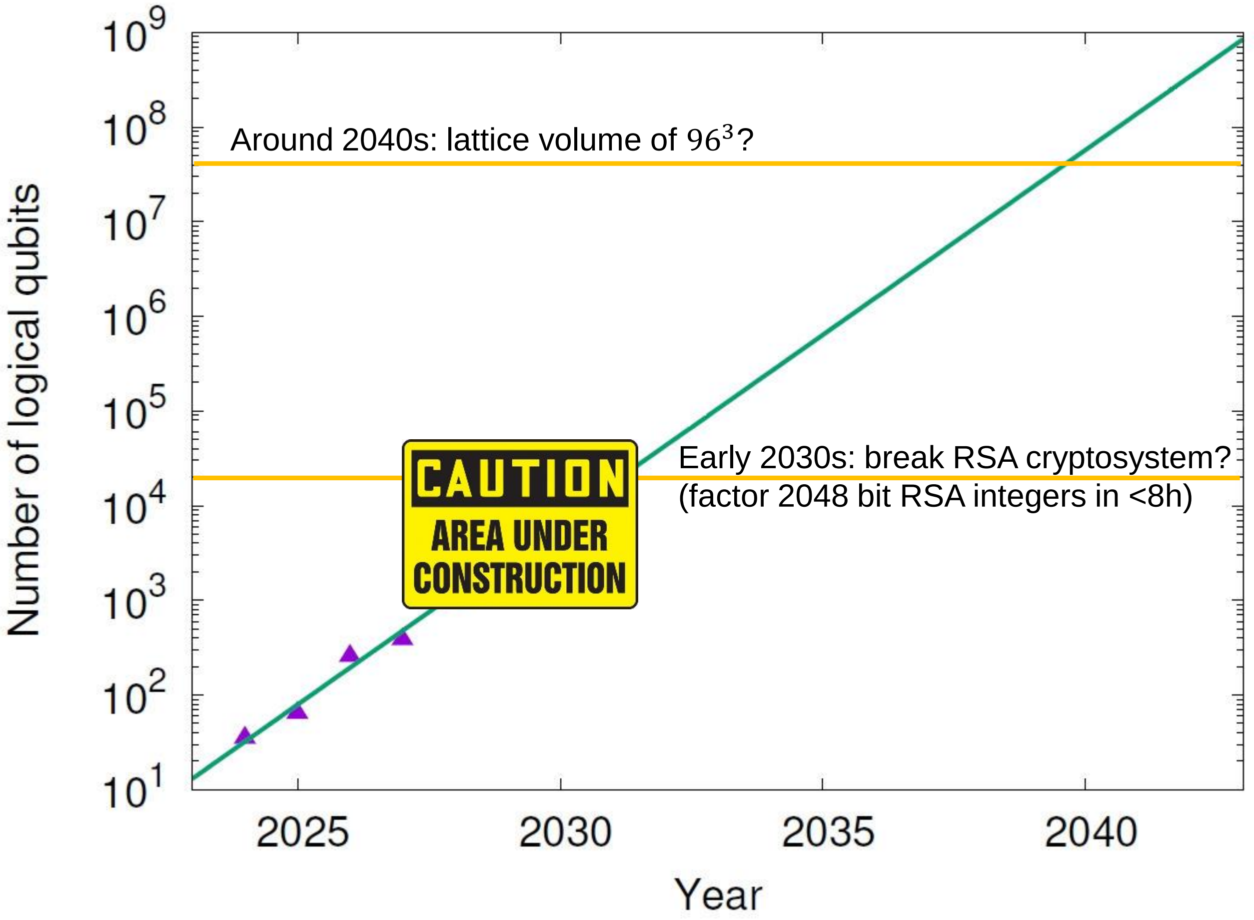
→ high risk, high gain

Classical

more established but limited

→ solid, promising





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

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

The Way Forward…

Quantum

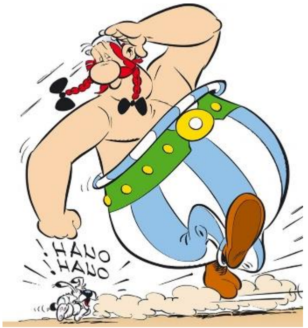
new and quickly progressing

→ high risk, high gain

Classical

more established but limited

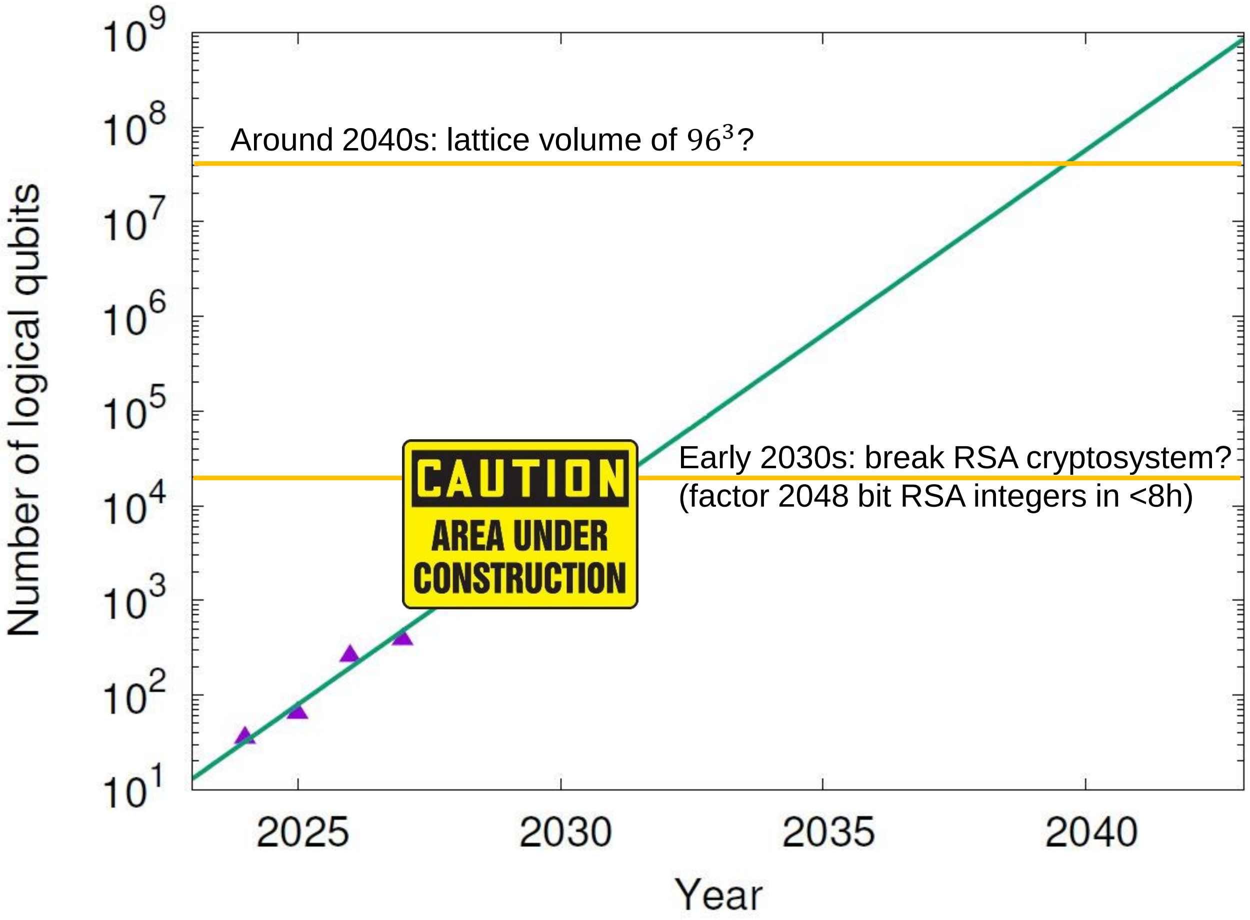
→ solid, promising



Combination

→ quantum-classical algorithms

A Rough Sketch…



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

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

The Way Forward…

Quantum

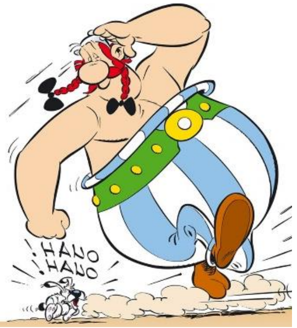
new and quickly progressing

→ high risk, high gain

Classical

more established but limited

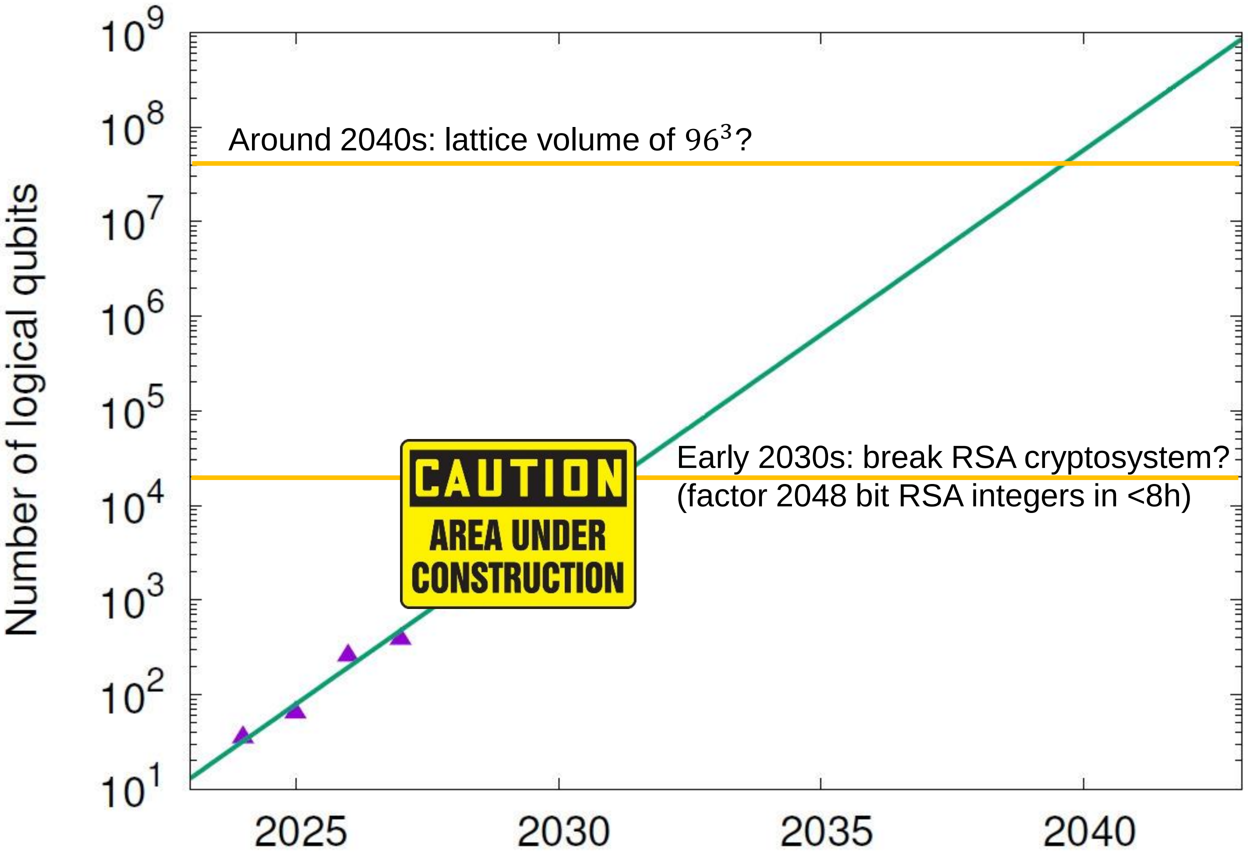
→ solid, promising



Combination

→ quantum-classical algorithms

A Rough Sketch…



Thank you for listening!

Do you have any questions?

Backup: details of Bose-Hubbard quantum simulation

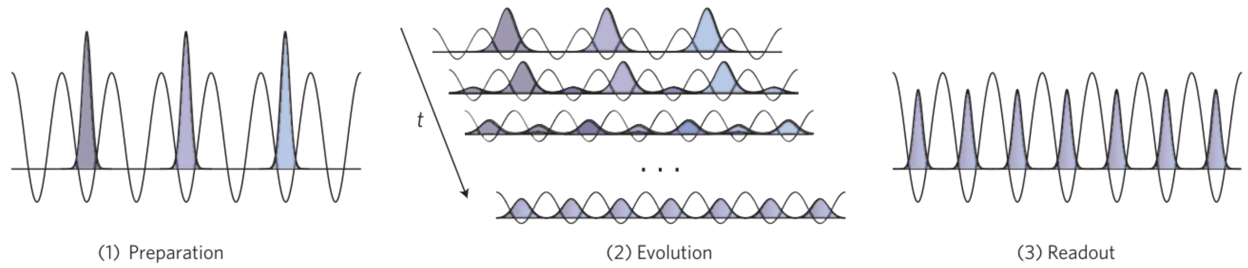
Classical versus quantum simulation | **Why is(n't) classical computing enough?**

Bose-Hubbard Hamiltonian

$$\mathcal{H} = \sum_j -J\left(\hat{a}_j^\dagger \hat{a}_{j+1} + h.c.\right) + \frac{U}{2}\hat{n}_j(\hat{n}_j - 1) + \frac{K}{2}\hat{n}_j j^2$$

Experimental goal [1]

Simulate quantum *tunneling* from even to odd sites



Experimental setup

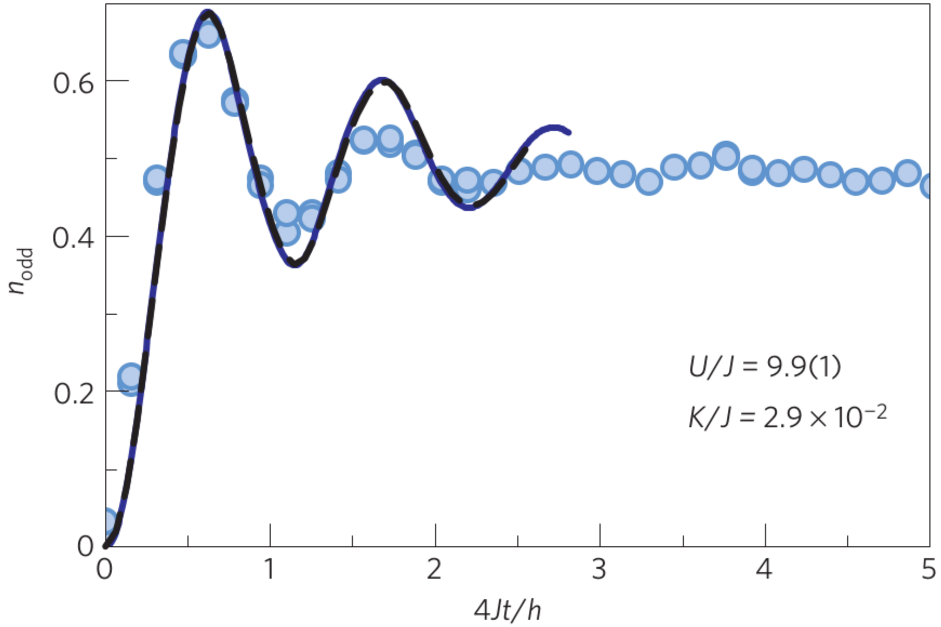
Quantum simulation: ultracold atoms in optical lattice

Classical benchmark: tensor networks (MPS-based)

[1] Trotzky et al. (2012)

Experimental results

Classical (line) vs. quantum (circles) simulation



this dynamical quantum simulator outperforms any continuous-time numerical simulation, for which the calculational effort increases with t. Simulation methods on classical computers, such as matrix-product state based t-DMRG used here, suffer from an extensive increase in entanglement entropy which limits the relaxation times accessible in the calculations[24,25].

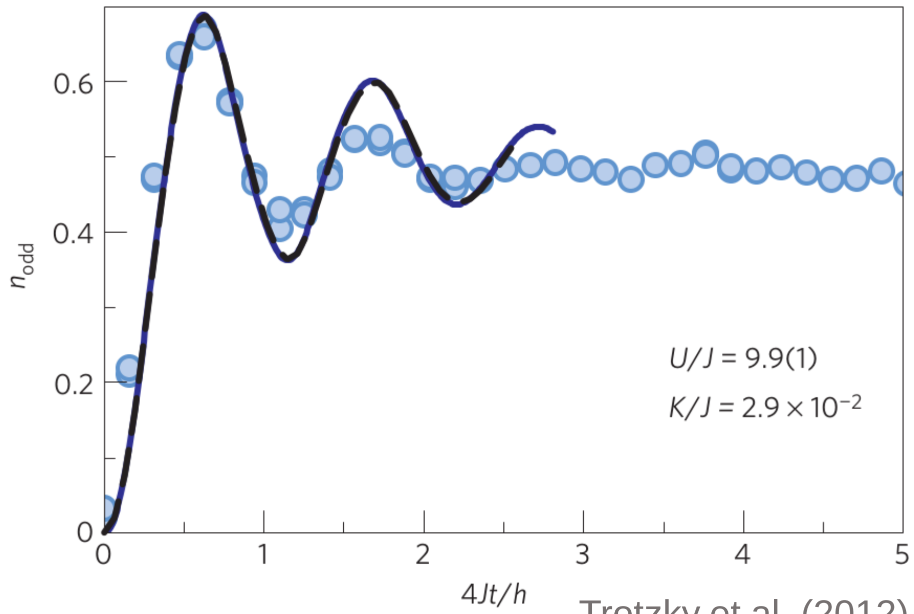
Backup: analog versus digital quantum computers

Analog	Digital

Analog

Concept

- Use controllable quantum system to simulate the behavior of another quantum system
- Continuous time evolution
- Usually non-universal

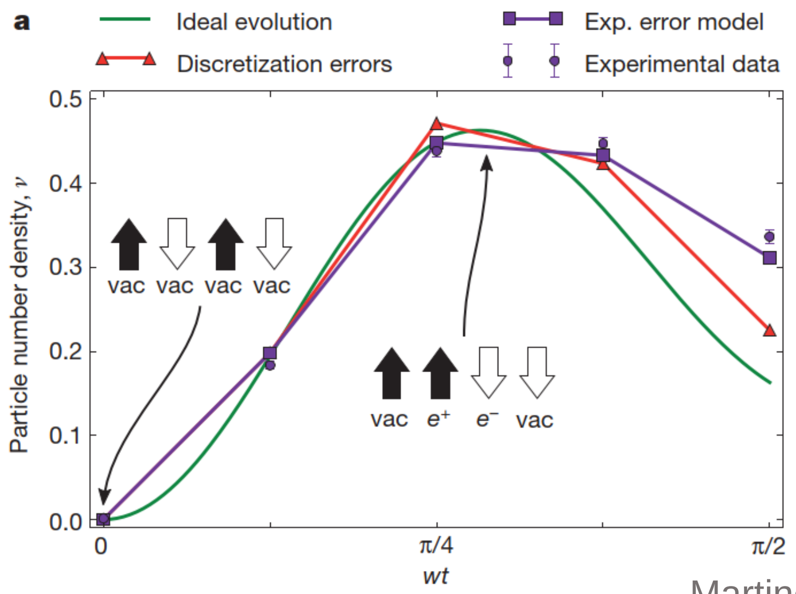


Trotzky et al. (2012)

Digital

Concept

- Construct set of logical gates onto qubits
- Discrete time evolution
- Usually universal



Martinez et al. (2016)

Backup: how to measure the energy in VQE?

Example: massless Schwinger model

Original Hamiltonian

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

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

Eliminate $\boldsymbol{\theta_n}$

$\phi_n^\dagger e^{i\theta_n}\phi_{n+1} \to \phi_n^\dagger \phi_{n+1}$ from gauge transformation:

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

Eliminate $\boldsymbol{F_n}$

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

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

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

Mapping the model to qubits

Dimensionless spin Hamiltonian [1]

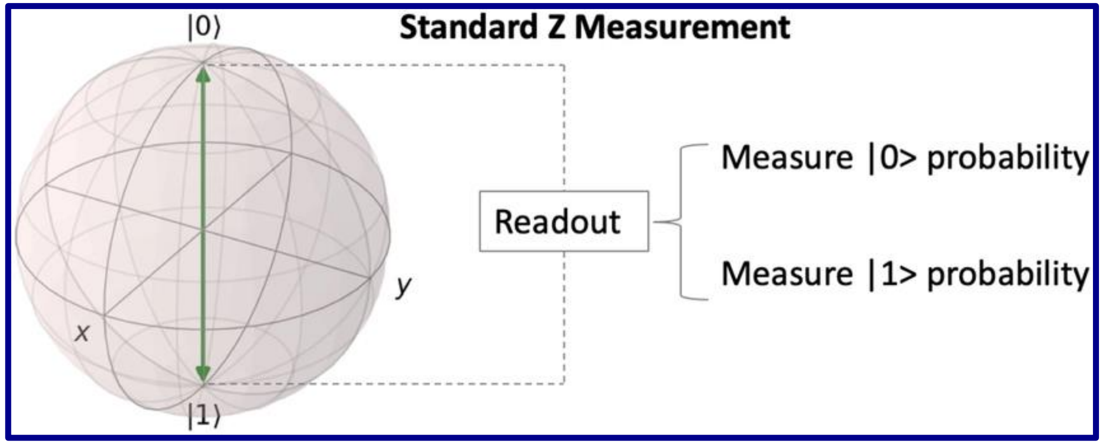
$$\mathcal{H} = x\sum_{n=0}^{N-2}\left(\sigma_n^+\sigma_{n+1}^- + \sigma_n^-\sigma_{n+1}^+\right) + \frac{1}{2}\sum_{n=0}^{N-2}\left\{\sum_{k=0}^{n}\left[(-1)^k + \sigma_k^z\right]\right\}^2$$

from mapping $\phi_n^\dagger\phi_{n+1} \to \sigma_n^+\sigma_{n+1}^-$ 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 $\boldsymbol{O} \in \{\mathbb{I}, \sigma^z\}^{\otimes N}$

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



Gokhale et al. (2020)

Backup: quantum volume

Concept	Timeline

Motivation

Number of *noisy* qubits: no good performance measure

New performance measure

Measure capabilities and error rates of quantum device

IBM's definition

$$\log_2 V_Q = \arg\max_{n \leq N}\{\min[n, d(n)]\}$$

Example

Successfully run circuit of depth $d = 8$ on $n = 8$ qubits: quantum volume is $V_Q = 2^8 = 256 \rightarrow$ size of state space

"Success"

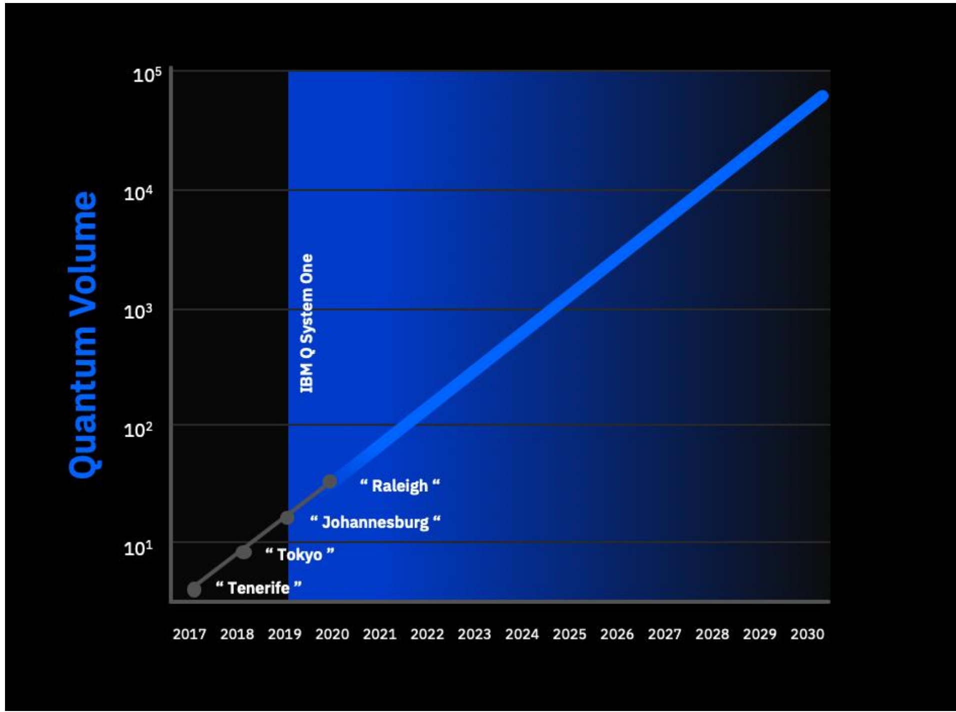
Most likely outputs of the circuit are computed correctly 67% of the time with a 2σ confidence interval

Last three years

Early 2020: $V_Q = 32$ (IBM) for $d = 5$, $n = 5$

Early 2021: $V_Q = 512$ (Honeywell) for $d = 9$, $n = 9$

Early 2022: $V_Q = 4096$ (Quantinuum) for $d = 12$, $n = 12$



Chow, Gambetta (2020)

Backup: preparing for overcoming sign problems in 3+1D

Example: 3+1D compact U(1) theory with θ-term

Goal

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

Analytical results

Derivation of Hamiltonian lattice θ-term:

$$\theta Q = -\frac{ig^2\theta}{8\pi^2 a} \sum_{n,i,j,k,b} \varepsilon_{ijk} \mathrm{Tr}\left[E^b_{n,i}\lambda^b \left(U_{n,jk} - U^\dagger_{n,jk}\right)\right]$$

Numerical results

Unlike in QCD, transition in U(1) might be not of first order

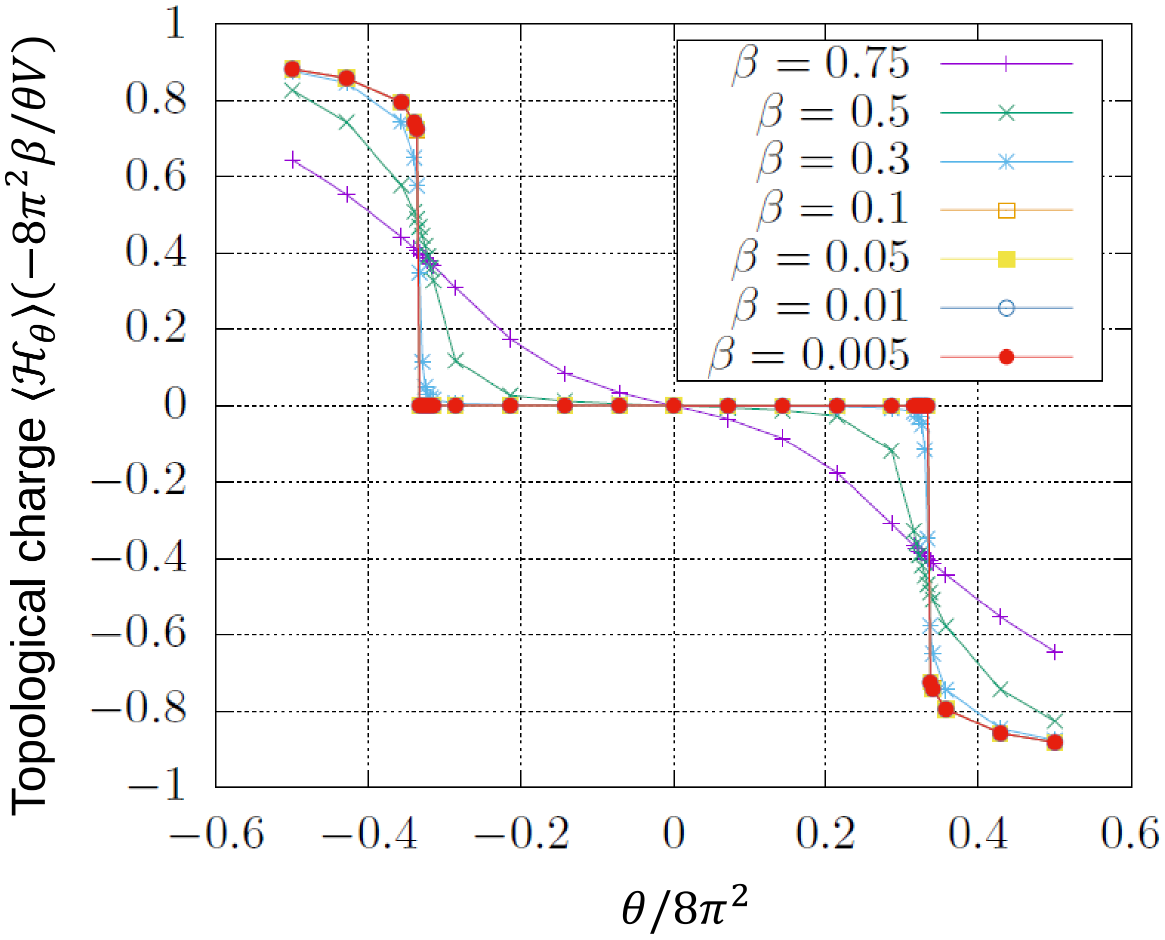
Near-future outlook

Larger-volume simulation with 3+1D tensor networks

Far-future outlook

First quantum simulation of 3+1D θ-term

Details: numerical ED results for single cube



Kan, LF, Kühn, Dellantonio, Zhang, Haase, Muschik, Jansen (2021a,2021b)