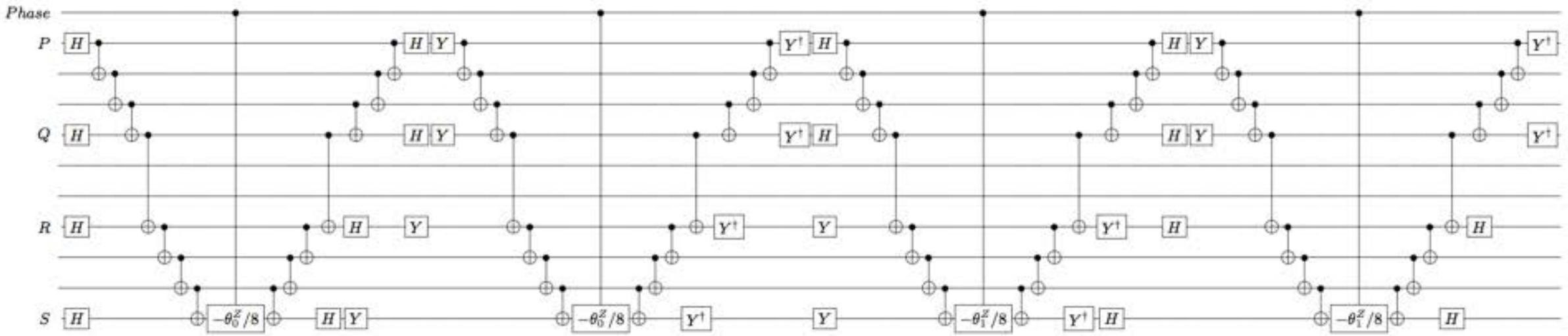


High performance quantum computing or how to program the future quantum computer

Matthias Troyer

Classical computers have come a long way

Antikythera mechanism

astronomical positions

(100 BC)

Kelvin's harmonic analyzer

prediction of tides

(1878)

digital devices

more complex

but general purpose

and error correcting

ENIAC

(1946)

天河-2

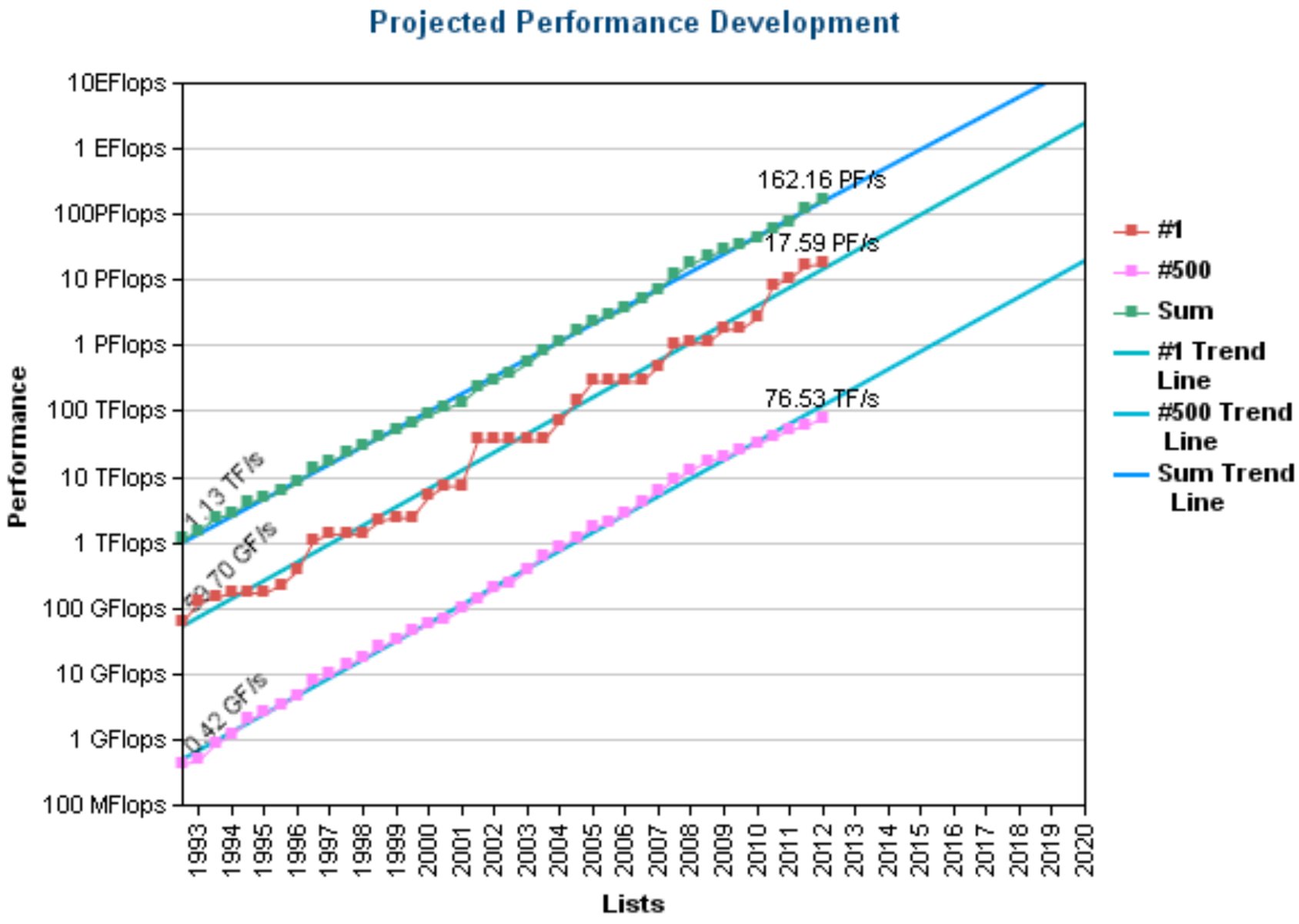
(2013)

analog devices

conceptually simple

calibration and scaling problems

The success of digital computers and Moore's law

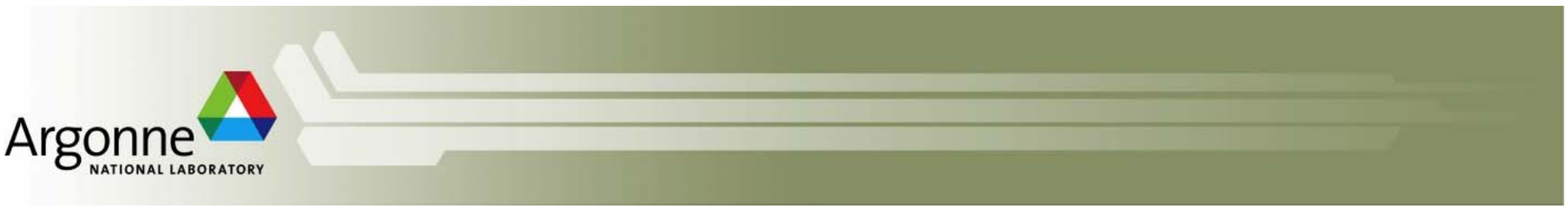


CPU cores:	1024	8'192	3'120'000	Amdahl's law: 99.99993% parallel
Power:	1 MW	3 MW	20 MW	1 MW = 1 MCHF / year

HPC 2014

HIGH PERFORMANCE COMPUTING

FROM CLOUDS AND BIG DATA TO EXASCALE AND BEYOND



Enabling technologies for beyond exascale computing

Paul Messina

Director of Science

Argonne Leadership Computing Facility

Argonne National Laboratory

July 9, 2014

Cetraro

What is “Beyond Exascale Computing?”

- **We are not referring to 10**21 flops**
- **“Beyond exascale” systems as we are defining them will be based on new technologies that will finally result in the much anticipated (but unknown) phase change to truly new paradigms/methodologies.**

We should not compare (potential) quantum computers to single cores of current CPUs

The competition are the best special purpose classical devices that one might build in 10-20 years

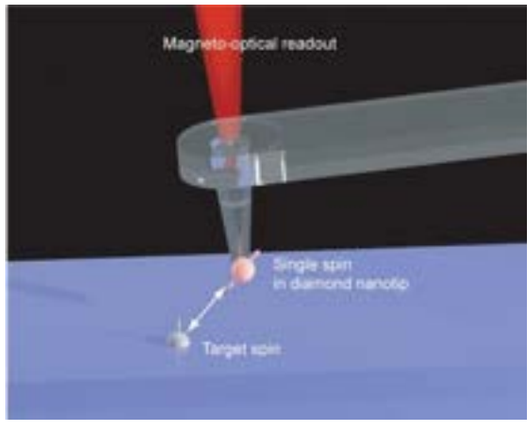
Beyond Moore's law: quantum devices

Quantum randomness

perfect random numbers

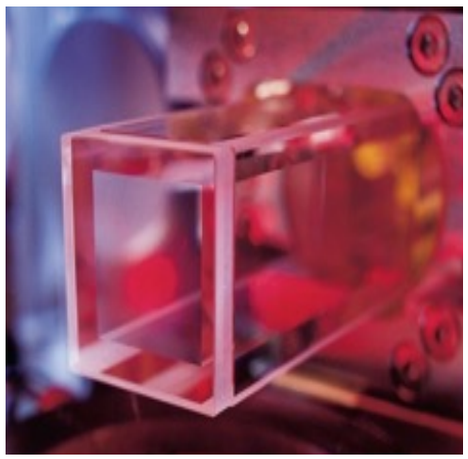
Quantum encryption

secure communication



Quantum sensing

high sensitivity sensors



Analog quantum simulators

solve quantum models

Quantum annealer

solve hard optimization problems?

Quantum computers?

Simulating quantum computers

- Need 2^N complex numbers to store the wave function of N qubits
- $O(2^N)$ classical operations to perform a quantum gate on N qubits

Qubits	Memory	Time for one gate
10	16 kByte	microseconds on a watch
20	16 MByte	milliseconds on smartphone
30	16 GByte	seconds on laptop
40	16 TByte	seconds on supercomputer
50	16 PByte	seconds on top supercomputer
60	16 EByte	minutes on future supercomputer
70	16 ZByte	hours on potential supercomputer?
…	…	…
250	size of visible universe	age of the universe

Step #0: the business case

Quantum computing beyond exa-scale

What are the important applications …

… that we can solve on a quantum computer …

… but not special purpose post-exa-scale classical hardware that we may build in ten years?

Microsoft IBM Google

NOKIA intel LOCKHEED MARTIN

What problems will we solve on a quantum computer?

design better drugs

cure cancer

counter climate change

fold proteins

fight hunger

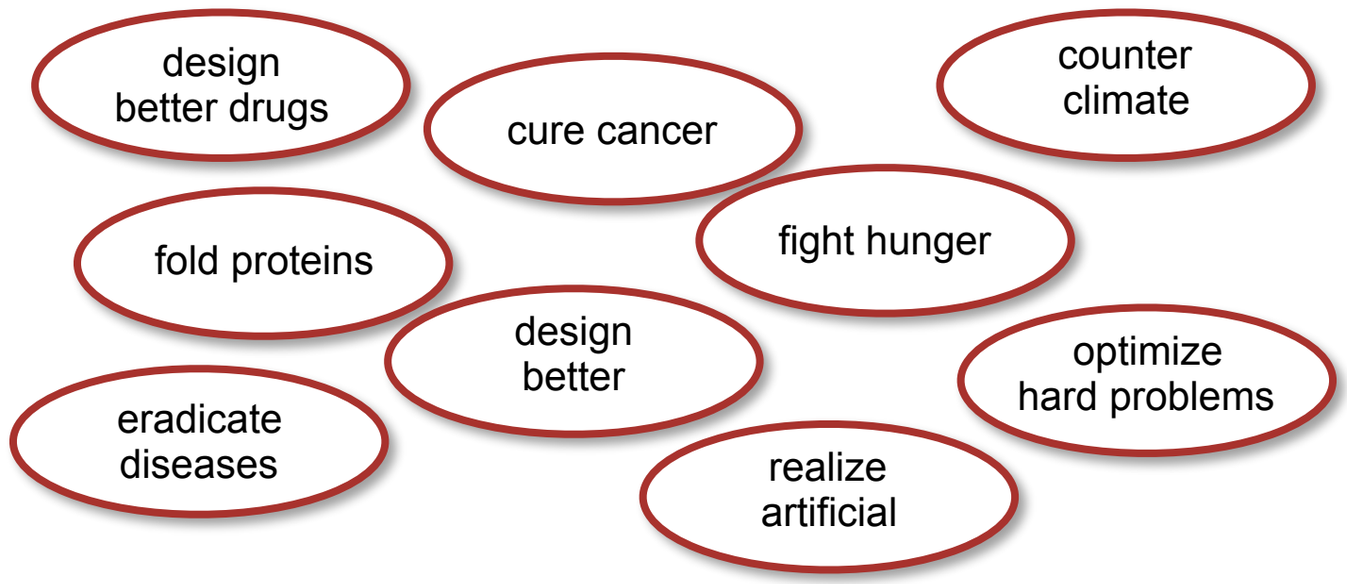
design better batteries

optimize hard problems

eradicate diseases

realize artificial intelligence

What problems will we solve on a quantum computer?



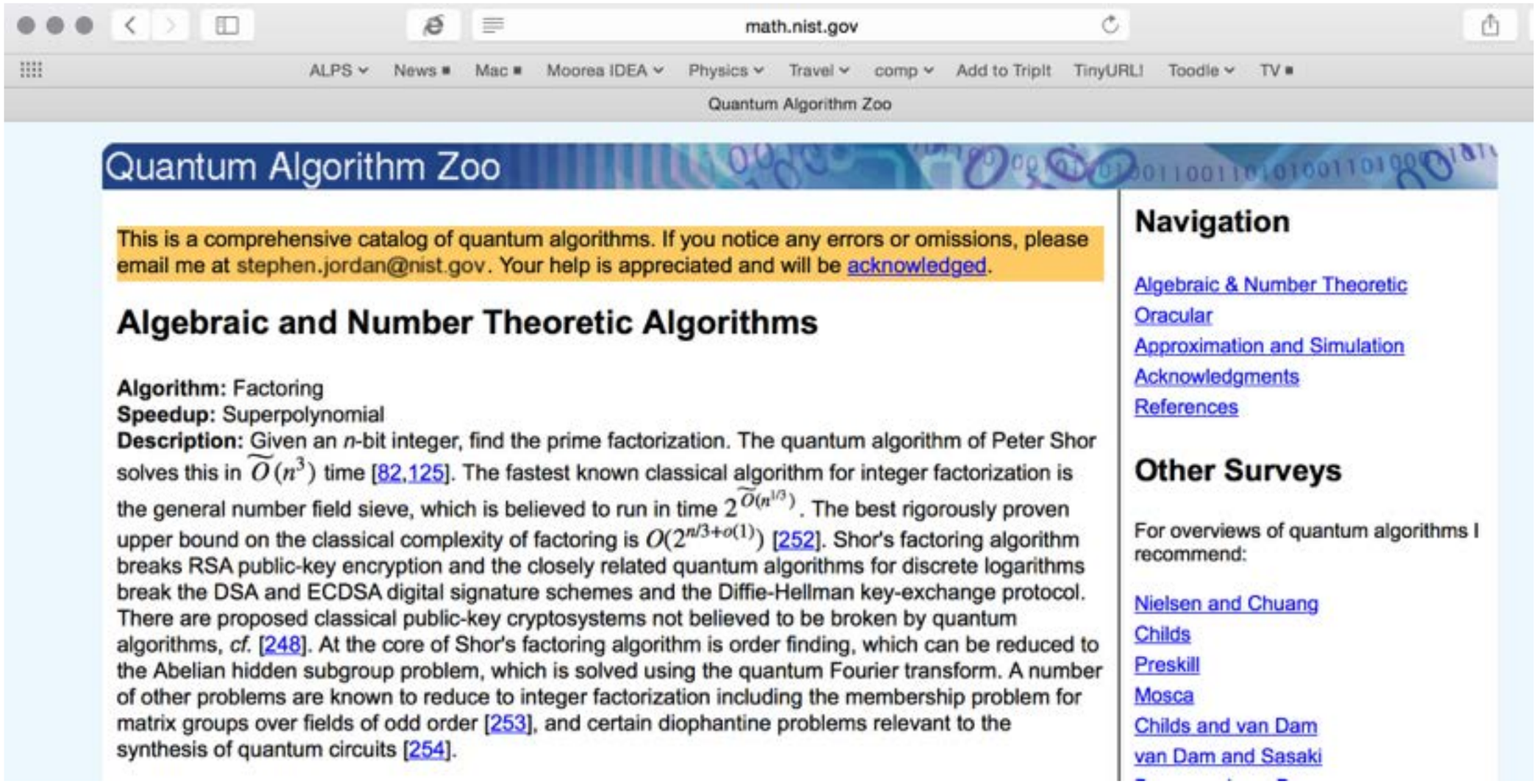
This is a list for a quantum wishing well

Which of these can actually profit from quantum computers?

Step #1: find a quantum algorithm

Many algorithms with known quantum speedup

Can we use any of them in real-world applications?



http://math.nist.gov/quantum/zoo/

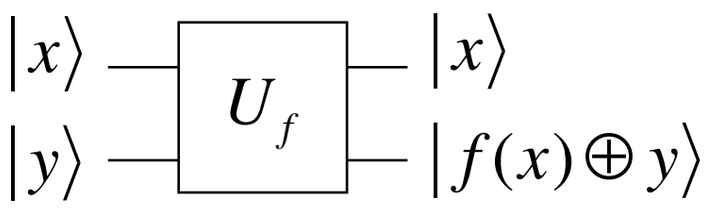
The Deutsch algorithm: simplest quantum speedup

- Check whether a binary function is constant or not

$$f:\{0,1\}\to\{0,1\}$$

- Classically two function calls are needed: f(0) = f(1)?

- Quantum mechanically only one function call by applying the function to both arguments at once

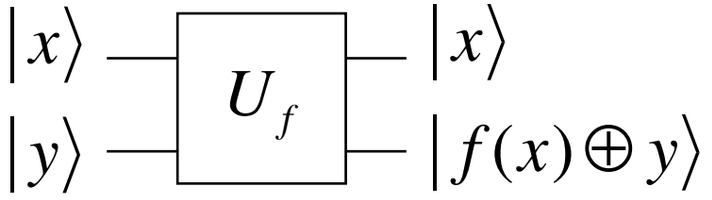


$$U_f|x\rangle|y\rangle\to|x\rangle|f(x)\oplus y\rangle$$

$$U_f\left(\alpha|0\rangle+\beta|1\rangle\right)|0\rangle\to\alpha|0\rangle|f(0)\rangle+\beta|1\rangle|f(1)\rangle$$

The Deutsch algorithm: simplest quantum speedup

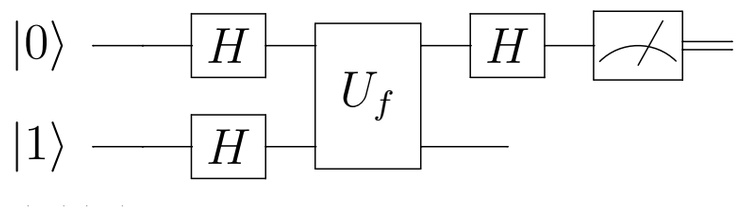
- Common misconception: quantum computers are faster since they work on all possible inputs in parallel



$$U_f|x\rangle|y\rangle \rightarrow |x\rangle|f(x)\oplus y\rangle$$

$$U_f\left(\alpha|0\rangle+\beta|1\rangle\right)|0\rangle \rightarrow \alpha|0\rangle|f(0)\rangle+\beta|1\rangle|f(1)\rangle$$

- If I measure the result I get **either** f(0) **or** f(1), chosen randomly!
- We need to compute one global result based on all inputs and measure just that one result!



$$\frac{1}{2}\left[\left(1+(-1)^{f(0)\oplus f(1)}\right)|0\rangle+\left(1-(-1)^{f(0)\oplus f(1)}\right)|1\rangle\right]$$

$$=\begin{cases} |0\rangle & \text{if } f(0)=f(1) \\ |1\rangle & \text{if } f(0)\neq f(1) \end{cases}$$

New computer architectures require new algorithms

- **Massively parallel computers**
  - use millions of loosely coupled CPU cores
  - we cannot just parallelize existing codes

- **Vector registers**
  - do operations on 8-16 numbers at once
  - need to rethink the program and change algorithms

- **GPUs**
  - very fast cores, all doing the same thing
  - need to rethink the program and change algorithms

- One cannot just translate existing algorithms
- Hardware-software co-design is needed to realize the potential of new architectures

Shor's algorithm for factoring

- Factoring is hard classically: O(exp($N^{1/3}$)) time for N bit - numbers

5369396836426911946079505415332600518604181838930231166202317318847061358416977798124777555435596464904452615804209177029240538156141035272554197625377862483029051809615050127043414927261020411423649694630967091077171430279795022115120241679622849447805650987368350247829683054309216276674509735105639240298977591783205062161915884859331945476609848287512883478098897975108372321438198667838135056716743636376259314981677010612529720589301303706515881099466219525234349036065726516132873421237667900245913537253744354928238018040554845306796065865605354860834270732796989421041371044010901319172800167312304864190643502624350075219901117888161765815866834760391595323095097926967071762530052007668467350605879541695798973080376300970096911310297914332946223591672260748684867072852791450573861929159507960

- But is polynomial time on a quantum computer
  - O(N^3) using minimal number of $2N$+3 qubits
  - O(N^2) using O(N) qubits
  - O(N) using O(N^2) qubits

- Shor's algorithm suddenly made quantum computing interesting

Shor's algorithm and encryption

- Shor's algorithm can be used to crack RSA encryption
  - assuming 10 ns gate time and minimal number of 2N+3 qubits
  - much faster (seconds) when using more qubits

RSA	cracked in	CPU years	Shor
453 bits	1999	10	1 hour
768 bits	2009	2000	5 hours
1024 bits		1000000	10 hours

- How does Shor's algorithm work?
  - It's not as simple as "trying all factors in parallel"
  - We again need to obtain a single answer combining all possible inputs

A factoring algorithm

This is hard classically!

4. Otherwise find the period ***r***, of the function $f(x) = a^x \bmod N$, that is the smallest $r>0$ for which $a^{x+r} \bmod N = a^x \bmod N$

A quantum computer can try all periods and find the right one by a Fourier transform!

Shor's algorithm and encryption

RSA	cracked in	CPU years	Shor
453 bits	1999	10	1 hour
768 bits	2009	2000	5 hours
1024 bits		1000000	10 hours

- But use of quantum computers to crack RSA is limited since we can switch to post-quantum encryption
  - quantum cryptography
  - lattice based cryptography

Challenge #0: the business case

Choose your favorite problem

design better drugs

cure cancer

counter climate change

fold proteins

fight hunger

design better batteries

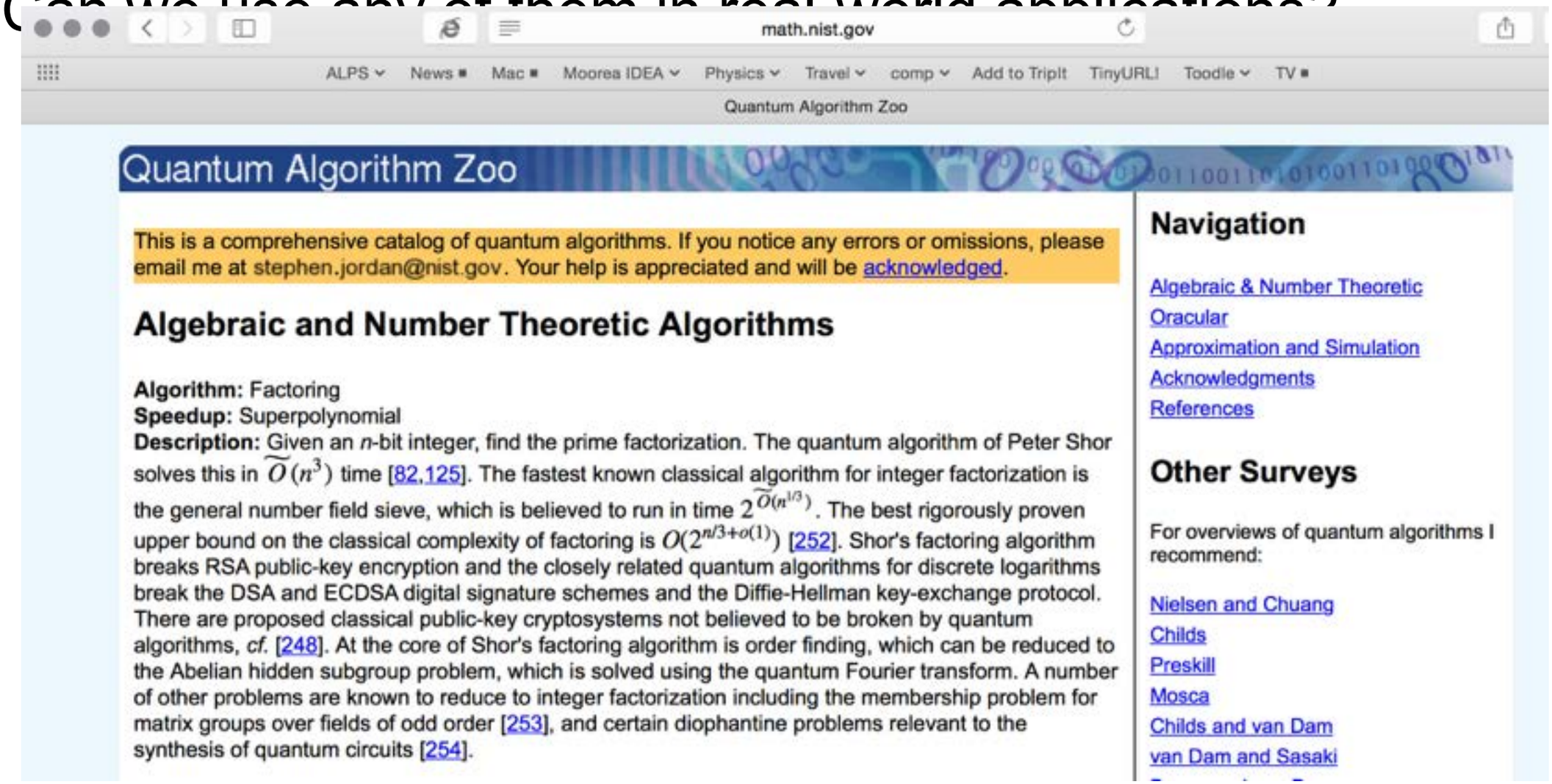
optimize hard problems

eradicate diseases

realize artificial intelligence

Find a matching quantum algorithm

Can we use any of them in real world applications?



http://math.nist.gov/quantum/zoo/

And develop it into a killer-app

7. Run the quantum algorithm to solve the problem

6. Embed into specific hardware and check runtime

5. Add error correction and check overhead and resources

4. Optimize code until logical circuit depth < 10^{14}

3. Check for quantum speedup

2. Implement all oracles and subroutines

1. Find quantum algorithm with quantum speedup

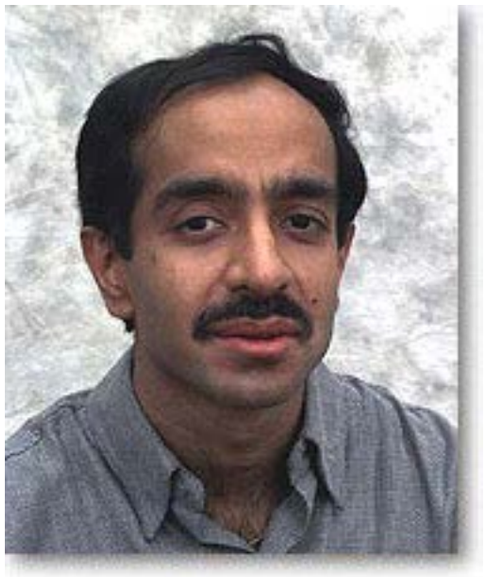
Step #2: implement the oracles

Step #3: check the speedup

Grover search: implementing the oracle

- Search an unsorted database of N entries in $\sqrt{N}$ time
- Rare case of provable quantum speedup given an oracle

- However, the oracle needs to be implemented!
  - N-entry database needs at least O(N) hardware resources to store the data
  - Can perform the same search classically in log(N) time with special purpose hardware



- Grover search is only useful if the database can be calculated on the fly
- Are there important real-world applications? Optimization problems?

Quantum page rank: checking the scaling

PRL 108, 230506 (2012) PHYSICAL REVIEW LETTERS week ending 8 JUNE 2012

Adiabatic Quantum Algorithm for Search Engine Ranking

Silvano Garnerone,[1,2,5] Paolo Zanardi,[2,5] and Daniel A. Lidar[2,3,4,5]

[1]*Institute for Quantum Computing, University of Waterloo, Waterloo, ON N2L 3G1, Canada*
[2]*Department of Physics & Astronomy, University of Southern California, Los Angeles, California 90089, USA*
[3]*Department of Electrical Engineering, University of Southern California, Los Angeles, California 90089, USA*
[4]*Department of Chemistry, University of Southern California, Los Angeles, California 90089, USA*
[5]*Center for Quantum Information Science & Technology, University of Southern California, Los Angeles, California 90089, USA*

(Received 25 October 2011; published 4 June 2012)

We propose an adiabatic quantum algorithm for generating a quantum pure state encoding of the PageRank vector, the most widely used tool in ranking the relative importance of internet pages. We present extensive numerical simulations which provide evidence that this algorithm can prepare the quantum PageRank state in a time which, on average, scales polylogarithmically in the number of web pages. We argue that the main topological feature of the underlying web graph allowing for such a scaling is the out-degree distribution. The top-ranked $\log(n)$ entries of the quantum PageRank state can then be estimated with a polynomial quantum speed-up. Moreover, the quantum PageRank state can be used in ''q-sampling'' protocols for testing properties of distributions, which require exponentially fewer measurements than all classical schemes designed for the same task. This can be used to decide whether to run a classical update of the PageRank.

Complexity of quantum page rank

- log(*N*) qubits are sufficient, but are not practical
- *N* qubits allow for a straightforward unary encoding

- Page rank then solved by adiabatic evolution of a spin model

$$H(s) = \sum_{i=1}^{n} h(s)_{ii} \sigma_i^+ \sigma_i^- + \sum_{i<j}^{n} h(s)_{ij} \left(\sigma_i^+ \sigma_j^- + \sigma_j^+ \sigma_i^- \right),$$

- Needs O(N^a) time with a ≈ 0.2…1
  when implementing N^2 couplings in parallel
  using O(N^2) hardware resources

- Classical page rank needs O(1) matrix-vector multiplications,
  each of complexity O(*dN*) where *d* is the mean number of links

Compare to classical hardware

	Quantum custom hardware	Classical general purpose	Classical custom hardware
Serial time complexity	$O(N^2 \text{ polylog}(N))$	$O(dN \log(N))$	$O(dN)$
Memory required	$O(N)$ qubits	$O(N)$ bits	$O(N)$ bits
Parallel time complexity	$O(N^{0.2}) – O(N)$	$O(N^{1/3})$ on 3D crossbar network	$O(\log(N))$
Hardware required	$O(N^2)$	$O(dN)$	$O(dN)$

- Quantum speedup vanishes when comparing parallel quantum hardware to parallel special purpose classical hardware
- Classical hardware requirements increase slower classically than quantum

Solving linear systems of equations

Harrow, Hassidim, Lloyd, PRL (2009)

- Solve linear system $Ax=b$ in $\log(N)$ time
- Requirements
  - Only $\log(N)$ bits of the answer are needed
  - Problem is well conditioned
  - Time evolution using the matrix A can be implemented efficiently

$$e^{-iAt}\left|b\right\rangle$$

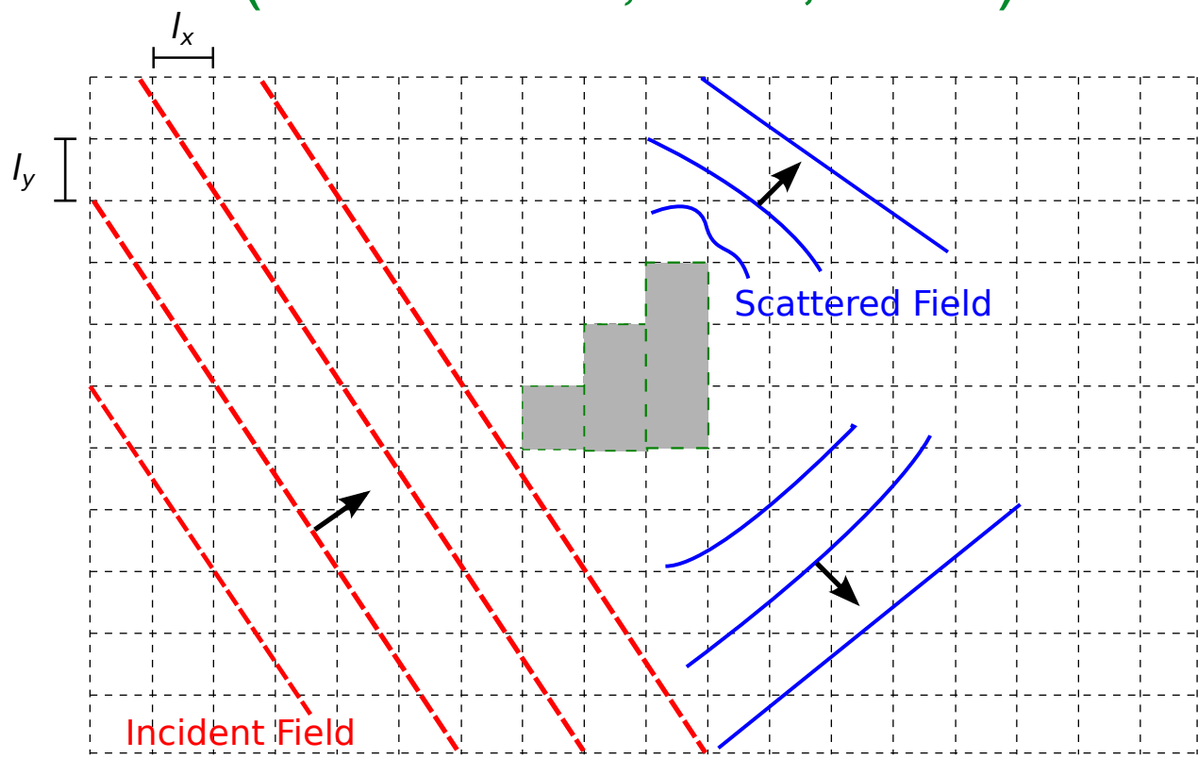
Implementing a general time evolution

- In general we always need O(N^2) gates if matrix A has N^2 different entries
- Lloyd, Mohseni and Rebentrost (Nature Physics 2014) propose an O(log(N)) implementation using QRAM to store square root of the matrix
  - needs O(N^2) gates and qubits to implement the QRAM
  - needs O(N^3) effort in classical preparation to calculate the square root
- How does it compare to classical approaches?
  - Gaussian elimination solves a linear system in O(N^3) time using O(N^2) memory
  - Using O(N^2) classical hardware we can do a matrix-vector multiplication in O(log(N)) time and solve the problem iteratively
- Furthermore, using O(N^2) classical hardware we can classically emulate any quantum algorithms acting on 2 log N qubits.

Solving linear systems of equations efficiently

- Quantum speedup can only be realized if the evolution exp(-*iAt*) can be implemented using a short circuit, i.e. it does not depend on lots of data

- Electromagnetic wave scattering problem (Clader *et al,* PRL, 2013)
  - finite difference discretization
  - represent shapes by splines



- Here is a definite exponential speedup

Challenge #4: check the runtime

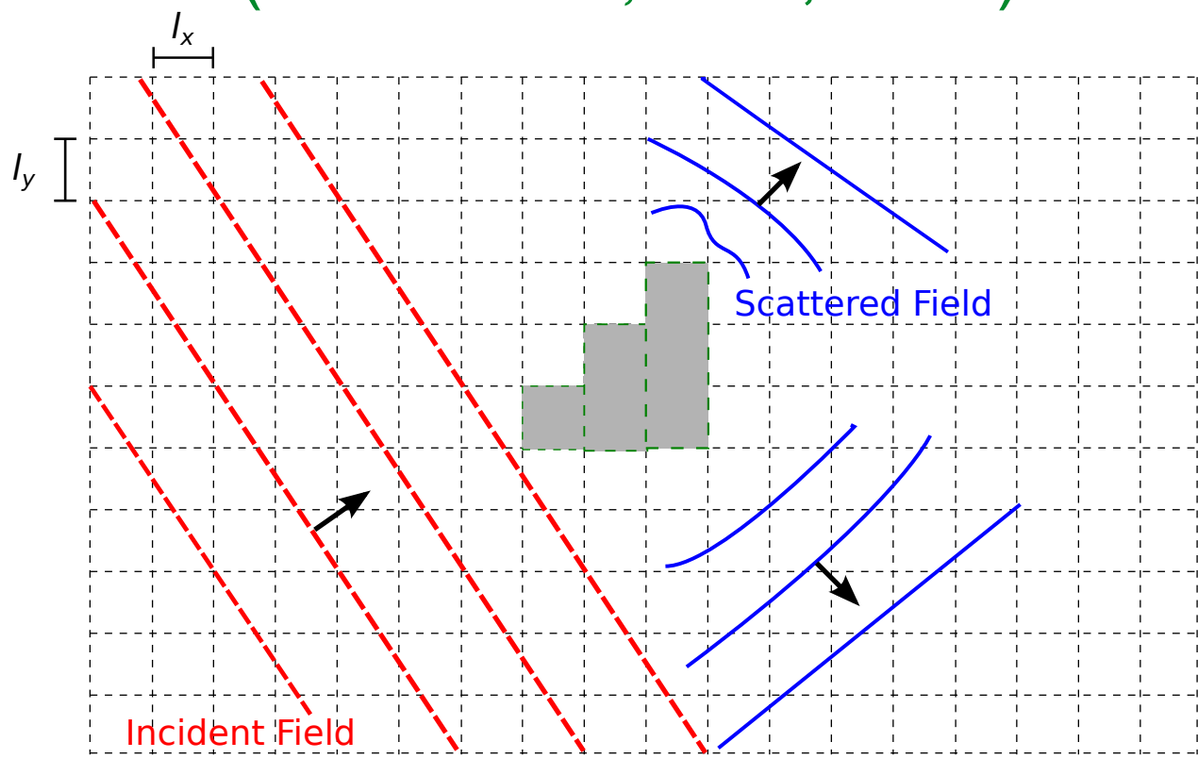
Now implement the quantum algorithm

- Use one of a number of quantum programming languages
  - QCL, one of the first quantum computer simulators
    http://tph.tuwien.ac.at/~oemer/qcl.html
  - Liquid (Microsoft), quantum computer simulation
    https://github.com/msr-quarc/liquid
  - Quipper, quantum compilation and resource estimation
    http://www.mathstat.dal.ca/~selinger/quipper/
  - ScaffCC, quantum compilation and resource estimation
    https://github.com/ajavadia/ScaffCC
  - OpenQu (our new project)
    - quantum compilation and optimization
    - optimized high-level and low-level quantum libraries
    - efficient simulation and emulation of 50+
    - interface to quantum hardware to run on actual devices (during 2016)

Solving linear systems of equations efficiently

- Quantum speedup can only be realized if the evolution exp(-*iAt*) can be implemented using a short circuit, i.e. it does not depend on lots of data

- Electromagnetic wave scattering problem (Clader *et al,* PRL, 2013)
  - finite difference discretization
  - represent shapes by splines

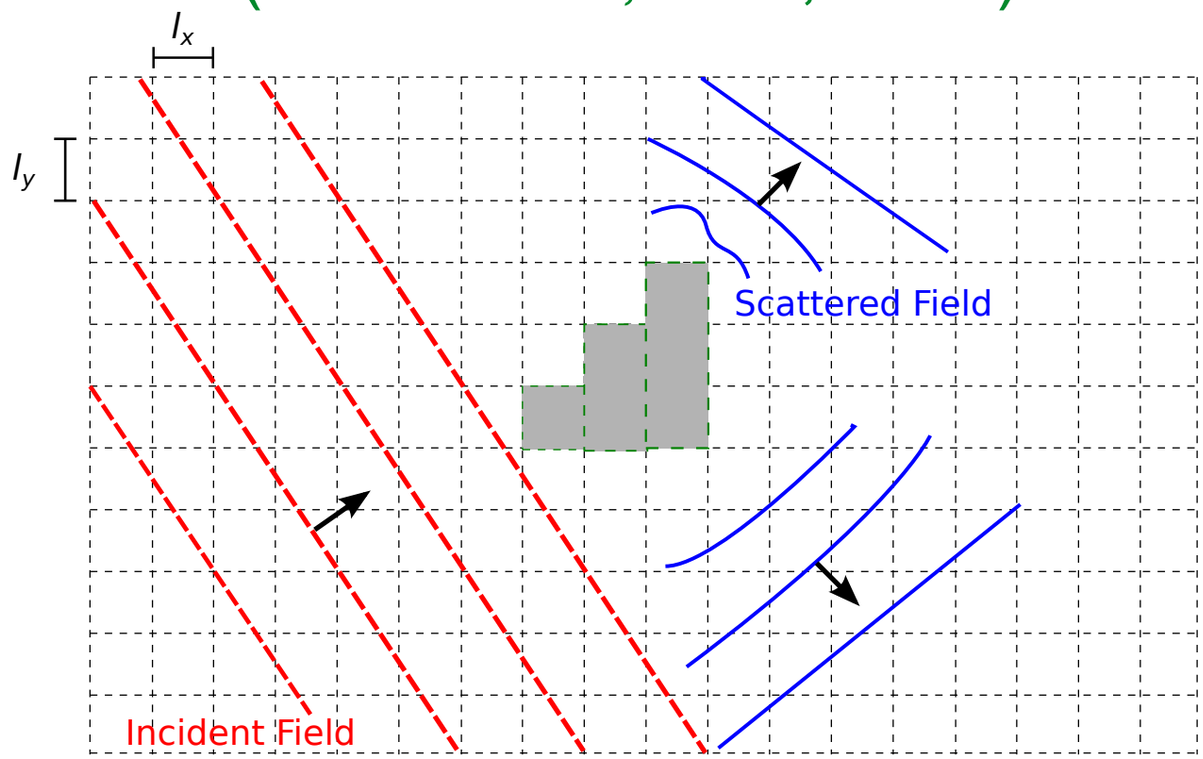


- Here is a definite exponential speedup

Solving linear systems of equations efficiently

- Quantum speedup can only be realized if the evolution exp(-*iAt*) can be implemented using a short circuit, i.e. it does not depend on lots of data

- Electromagnetic wave scattering problem (Clader *et al,* PRL, 2013)
  - finite difference discretization
  - represent shapes by splines



- Here is a definite exponential speedup
- But seems to require 10^{29} gate operations (see arXiv:1505.06552)?

Do we have a problem?

	Ion traps	Superconducting	Hypothetical target
Gate time	10 μs	10 ns	1 ns
Time for 10^{25} gates	$3x10^{12}$ a	$3x10^{9}$ a	$3x10^{8}$ a
Gates in two weeks runtime	10^{11}	10^{14}	10^{15}

A quantum algorithm should not just be of polynomial complexity

It needs to solve an interesting problem in less than 10^{15} operations

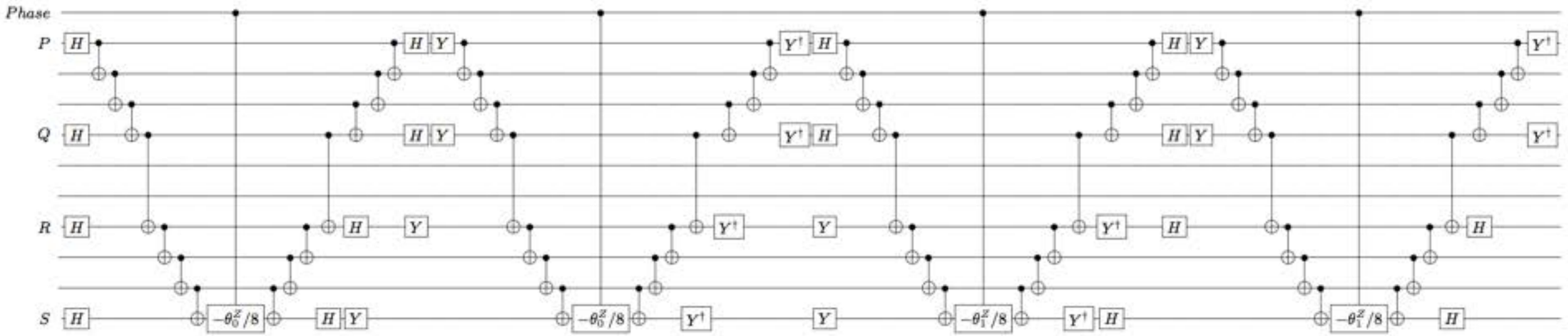
This means details of the gate set also matter

New computer architectures require new algorithms

- **Massively parallel computers**
  - use millions of loosely coupled CPU cores
  - we cannot just translate existing codes, they will not scale
- **Vector registers**
  - do operations on 8-16 numbers at once
  - need to rethink the program and change algorithms
- **GPUs**
  - very fast cores, all doing the same thing, on limited local memory
  - need to rethink the program and change algorithms to use those cores
- One cannot just translate existing algorithms but hardware-software co-design is needed to realize the potential of new architectures

Quantum software engineering

- **Polynomial scaling is not sufficient**
- **Hardware-software co-design**
  - We need to develop quantum software in parallel to quantum hardware, similar to classical supercomputers
- **Optimized algorithms**
  - Develop faster quantum algorithms
  - Rethink the solution of problems on quantum computers
- **Optimized libraries**
  - develop highly optimized quantum libraries for common operations instead of machine-translating classical codes
- **We have routinely accelerated classical codes by factors of more than 10^6 and and now need to do the same for quantum codes**



High performance quantum computing or how to program the quantum computer

Matthias Troyer

Quantum computing beyond exa-scale

What are the important applications …

… that we can solve on a quantum computer …

… but not special purpose post-exa-scale classical hardware that we may build in ten years?

And develop it into a killer-app

7. Run the quantum algorithm to solve the problem

6. Embed into specific hardware and check runtime

5. Add error correction and check overhead and resources

4. Optimize code until logical circuit depth < 10^{14}

3. Check for quantum speedup

2. Implement all oracles and subroutines

1. Find quantum algorithm with quantum speedup

First applications reaching a petaflop

Domain area	Code name	Institution	# of cores	Performance	Notes
Materials	DCA++	ORNL	213,120	**1.9 PF**	2008 Gordon Bell Prize Winner
Materials	WL-LSMS	ORNL/ETH	223,232	**1.8 PF**	2009 Gordon Bell Prize Winner
Chemistry	NWChem	PNNL/ORNL	224,196	1.4 PF	2008 Gordon Bell Prize Finalist
Materials	DRC	ETH/UTK	186,624	1.3 PF	2010 Gordon Bell Prize Hon. Mention
Nanoscience	OMEN	Duke	222,720	> 1 PF	2010 Gordon Bell Prize Finalist
Biomedical	MoBo	GaTech	196,608	780 TF	2010 Gordon Bell Prize Winner
Chemistry	MADNESS	UT/ORNL	140,000	550 TF	
Materials	LS3DF	LBL	147,456	442 TF	2008 Gordon Bell Prize Winner
Seismology	SPECFEM3D	USA (multiple)	149,784	165 TF	2008 Gordon Bell Prize Finalist

Source: T. Schulthess

Simulating quantum systems

Feynman invented quantum computers to simulate quantum physics

We can surpass the best classical computers with only a little more than 50 qubits!

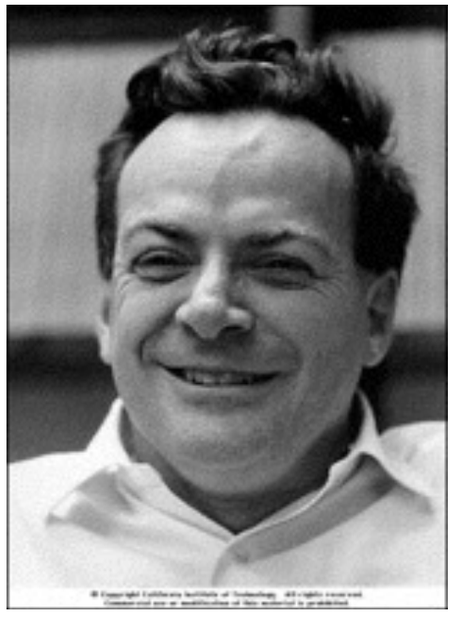
International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

Richard P. Feynman

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

Received May 7, 1981



Solving quantum chemistry on a quantum computer

- A killer-app for quantum computing is solving quantum problems
  - Design a room-temperature superconductor
  - Develop a catalyst for carbon sequestration

  - Develop better catalysts for nitrogen fixation (fertilizer)

- These problem need better accuracy than we get by using approximate classical algorithms
  - exponentially hard classically
  - polynomial complexity on quantum hardware

Preparing the ground state

- On a classical computer
  - Imaginary time projection
  - Power method or other iterative eigensolver

$$\left|\Psi_{GS}\right\rangle = \lim_{\tau\to\infty} e^{-\tau H}\left|\Psi_T\right\rangle$$

$$\left|\Psi_{GS}\right\rangle = \lim_{n\to\infty} (H-\Lambda)^n\left|\Psi_T\right\rangle$$

- On a quantum computer
  - ~~Imaginary time evolution~~
  - ~~Power method~~
  - Unitary operations + measurements:
    1. adiabatically prepare trial state
    2. projectively measure energy
    3. obtain the ground state if the ground state energy was measured

$$\left|\Psi_T\right\rangle$$

$$\left|\Psi_T\right\rangle \to \left|\phi_n\right\rangle \text{ with } H\left|\phi_n\right\rangle = E_n\left|\phi_n\right\rangle$$

$$\left|\phi_n\right\rangle \text{ picked with propability } \left|\left\langle\phi_n\middle|\Psi_T\right\rangle\right|^2$$

Quantum phase estimation

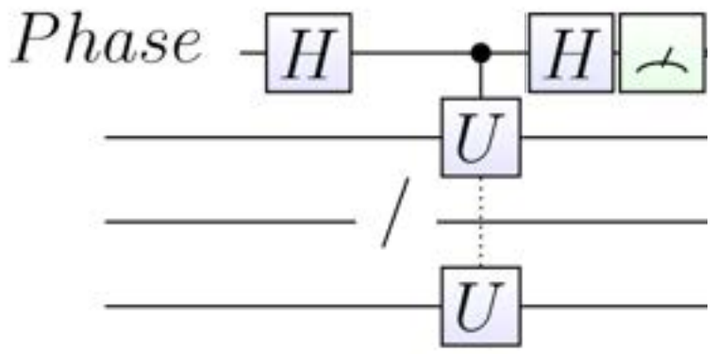
- Energy can be measured by measuring the phase of a wave function after unitary evolution

$$U\left|\phi_n\right\rangle \equiv e^{-iHt}\left|\phi_n\right\rangle = e^{-iE_n t}\left|\phi_n\right\rangle = e^{-i\phi}\left|\phi_n\right\rangle$$

- We can only measure relative phases, thus compare to the original phase by doing a controlled evolution

$$|0\rangle|\phi_n\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)|\phi_n\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle|\phi_n\rangle+U|1\rangle|\phi_n\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle+e^{-i\phi}|1\rangle\right)|\phi_n\rangle \rightarrow \frac{1}{2}\left((1+e^{-i\phi})|0\rangle+(1-e^{-i\phi})|1\rangle\right)|\phi_n\rangle$$

- Measure the ancilla phase qubit to obtain
  - 0 if the phase is 0 mod 2π
  - 1 if the phase is π mod 2π



- Repeat the procedure to measure the phase and energy more accurately

Solving quantum chemistry on a quantum computer

1. Select a finite (generally non-orthogonal) basis set

2. Perform a Hartree-Fock calculation to
   - get an approximate solution
   - get an orthogonal basis set

3. Find the true ground state of the Hamiltonian in this new basis set

$$H = \sum_{pq} t_{pq} c_p^\dagger c_q + \sum_{pqrs} V_{pqrs} c_p^\dagger c_q^\dagger c_r c_r$$

exact classical approach: full-configuration interaction
exponential complexity!

Whitfield, Biamonte, Aspuru-Guzik
Molecular Physics (2011)

Solving quantum chemistry on a quantum computer

1. Select a finite (generally non-orthogonal) basis set

2. Perform a Hartree-Fock calculation to
   - get an approximate solution
   - get an orthogonal basis set

3. Find the true ground state of the Hamiltonian in this new basis set

$$H = \sum_{pq} t_{pq} c_p^\dagger c_q + \sum_{pqrs} V_{pqrs} c_p^\dagger c_q^\dagger c_r c_r$$

4. Prepare a good guess for the ground state

5. Perform quantum phase estimation to
   - get the ground state wave function
   - get the ground state energy

Whitfield, Biamonte, Aspuru-Guzik
Molecular Physics (2011)

Time evolution under the Coulomb Hamiltonian

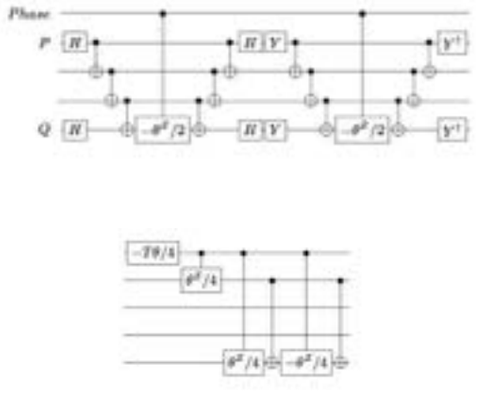
- Key ingredient: evolve the wave function under the Coulomb Hamiltonian

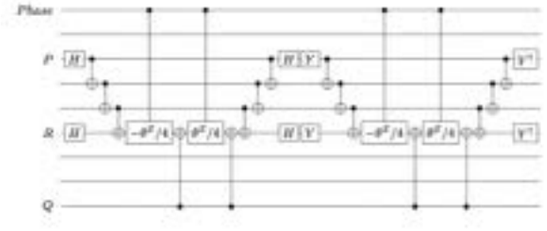
$$H = \sum_{pq} t_{pq} c_p^\dagger c_q + \sum_{pqrs} V_{pqrs} c_p^\dagger c_q^\dagger c_r c_r \equiv \sum_{m=1}^{M} H_m \qquad M = O(N^4) \text{ terms}$$

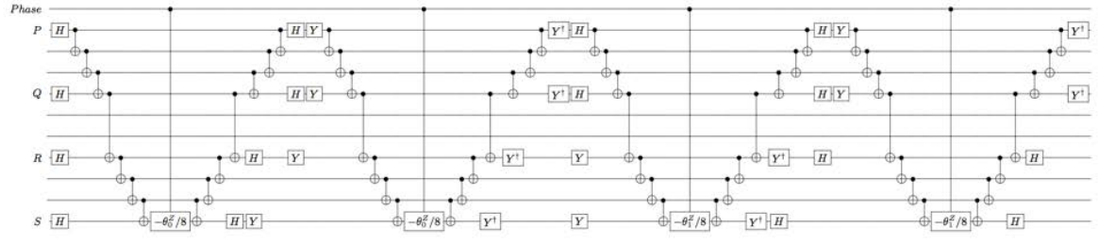
- Use Trotter breakup to implement time evolution
  Whitfield, Biamonte, Aspuru-Guzik, Molecular Physics (2011)

$$e^{-\Delta t H} \approx \prod_{m=1}^{M} e^{-\Delta \tau H_m}$$

- Efficient circuits available for each of the $M=N^4$ terms







Representing fermion terms by quantum circuits

- Map the occupation of each spin-orbital to the states of one qubit

$$|0\rangle = |\uparrow\rangle \qquad |1\rangle = |\downarrow\rangle$$

- Density operators get mapped to Pauli matrices

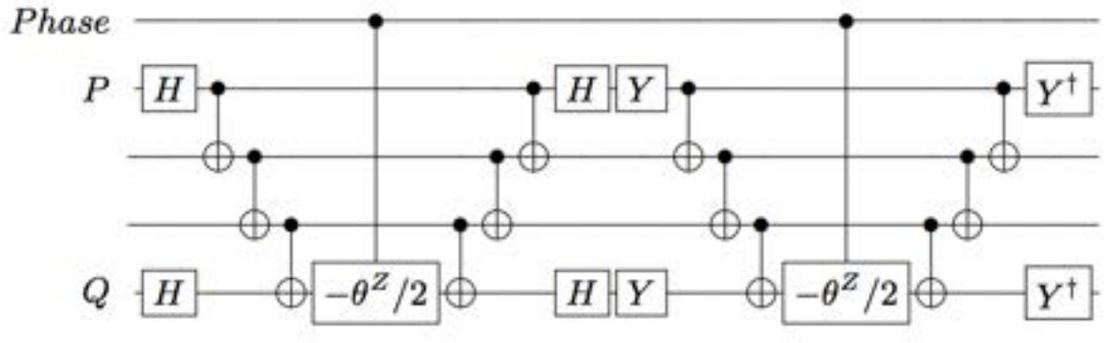
$$n_i = \frac{1}{2}\left(1 - \sigma_i^z\right)$$

- Hopping terms get mapped to spin flips with Jordan-Wigner strings

$$c_p^\dagger c_q = \sigma_p^- \prod_{i=p+1}^{q-1} \sigma_i^z \sigma_p^+$$

- Time evolution gets mapped to circuits built from unitary gates

$$e^{-i\theta\left(c_p^\dagger c_q + c_q^\dagger c_p\right)} =$$



Can quantum chemistry be performed on a small quantum computer?

Dave Wecker,[1] Bela Bauer,[2] Bryan K. Clark,[2,3] Matthew B. Hastings,[2,1] and Matthias Troyer[4]

[1] *Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA 98052, USA*
[2] *Station Q, Microsoft Research, Santa Barbara, CA 93106-6105, USA*
[3] *Kavli Institute for Theoretical Physics, University of California, Santa Barbara, CA 93106, USA*
[4] *Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland*

Phys. Rev. A **90**, 022305 (2014)

Can a classically-intractable problem be solved on a small quantum computer?

Can a classically-intractable problem be solved on a huge quantum computer?

Can a classically-intractable problem be solved on the largest imaginable quantum computer?

Solving a small non-trivial problem

- How many spin-orbitals do we need for an interesting problem?

Cuprate high-T_c superconductor	O(10000)
Active space of interesting reactions	200-400
Classically tractable by DMRG	≈ 70
Classically tractable by full-CI	≈ 50

- What accuracy do we need?

Wish of my chemistry colleague	0.1mHa - 1μHa
Modest goal	**1mHa**

Note that the total energies are of the order of 1kHa

We thus need at least six digits of precision

Our initial outlook in January 2014

- Second order Trotter formula for m terms needs $m^{3/2}$ Trotter steps per time interval

$$5^{2k} m^2 ||H|| t \left(\frac{m||H||t}{\epsilon} \right)^{1/2k}$$

- Naïve scaling with number of spin-orbitals

	N_t	$\frac{\text{Gates}}{\text{term}}$	$1/\Delta_t$	Total (Parallel)
Upper bound	N^4	N	$(N_t)^{3/2} = N^6$	N^{11} (N^{10})
Empirical scaling	$N^{3.8}$	N	N^4	N^9 (N^8)

- Estimates for an example molecule: Fe_2S_2

Gate count	10^{18}
Parallel circuit depth	10^{17}
Run time @ 10ns gate time	30 years

Improvements to the quantum algorithms:

- Estimates for an example molecule: Fe_2S_2 with 118 spin-orbitals

Gate count	10^{18}
Parallel circuit depth	10^{17}
Run time @ 10ns gate time	30 years

New gate count	10^{11}
Parallel circuit depth	10^{10}
Run time @ 10ns gate time	2 minutes

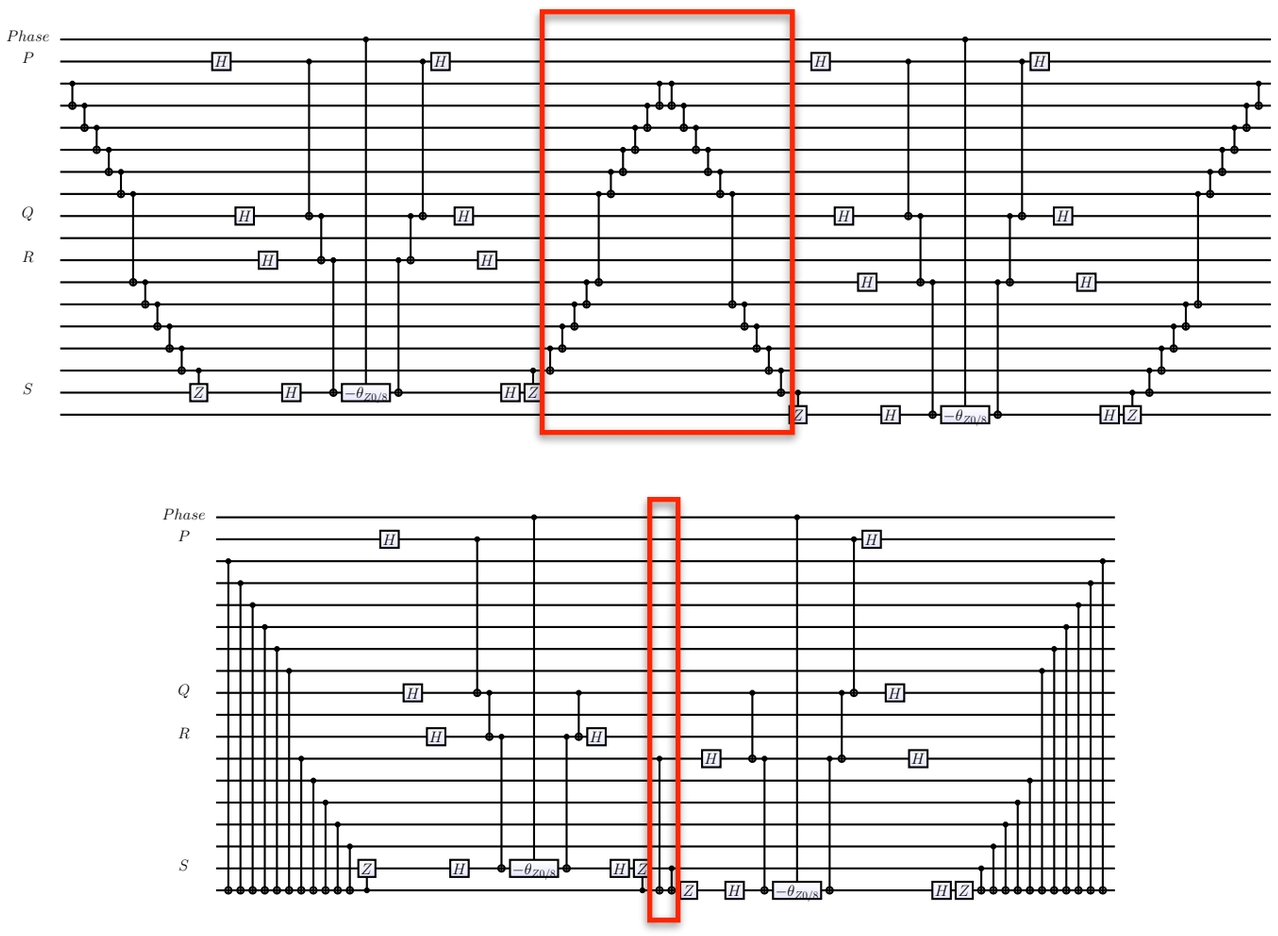
- Attempting to reduce the horrendous runtime estimates we achieved
  Wecker *et al.*, PRA (2014), Hastings *et al.*, QIC (2015), Poulin *et al.*, QIC (2015)

Cancelling of Jordan-Wigner strings:	O(*N*) reduction in gates
Nesting of terms:	O(*N*) reduction in circuit depth
Optimizing circuits:	4x reduction in gates
Smart interleaving of terms:	10x reduction in Trotter steps
Multi-resolution Trotter:	10x reduction in gates
Better quantum phase estimation:	4x reduction in rotation gates

Optimization 1: reducing Jordan-Wigner strings

M.B. Hastings *et* al., QIC (2015)

- Rearrange the circuits and use an optimized ordering of terms to cancel most of the Jordan-Wigner strings between terms

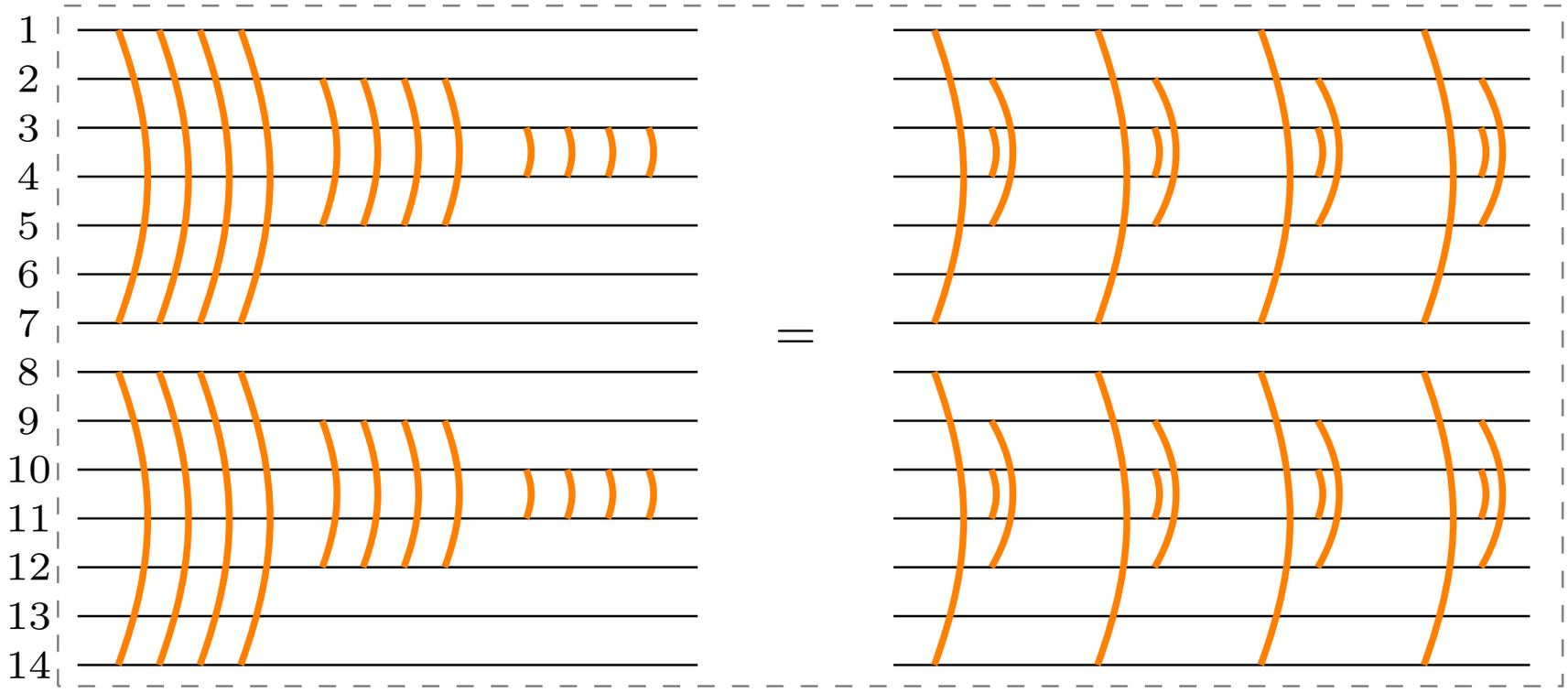


- Reduces the cost of Jordan-Wigner strings from O(*N*) to amortized O(1)

Optimization 2: Nesting of terms

M.B. Hastings *et* al., QIC (2015)

- Interleave terms that seem to interfere at first sight due to overlapping Jordan-Wigner strings



- Can do O(*N*) terms in parallel for another power reduction

Optimization 3: faster phase estimation

- **Standard phase estimation**
  - propagates by time *t*
  - controlled rotation gates to implement evolution (need two normal rotations)

$$\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)|\phi_n\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle|\phi_n\rangle+|1\rangle e^{-iHt}|\phi_n\rangle\right)=\frac{1}{\sqrt{2}}\left(|0\rangle+e^{-i\phi}|1\rangle\right)|\phi_n\rangle$$

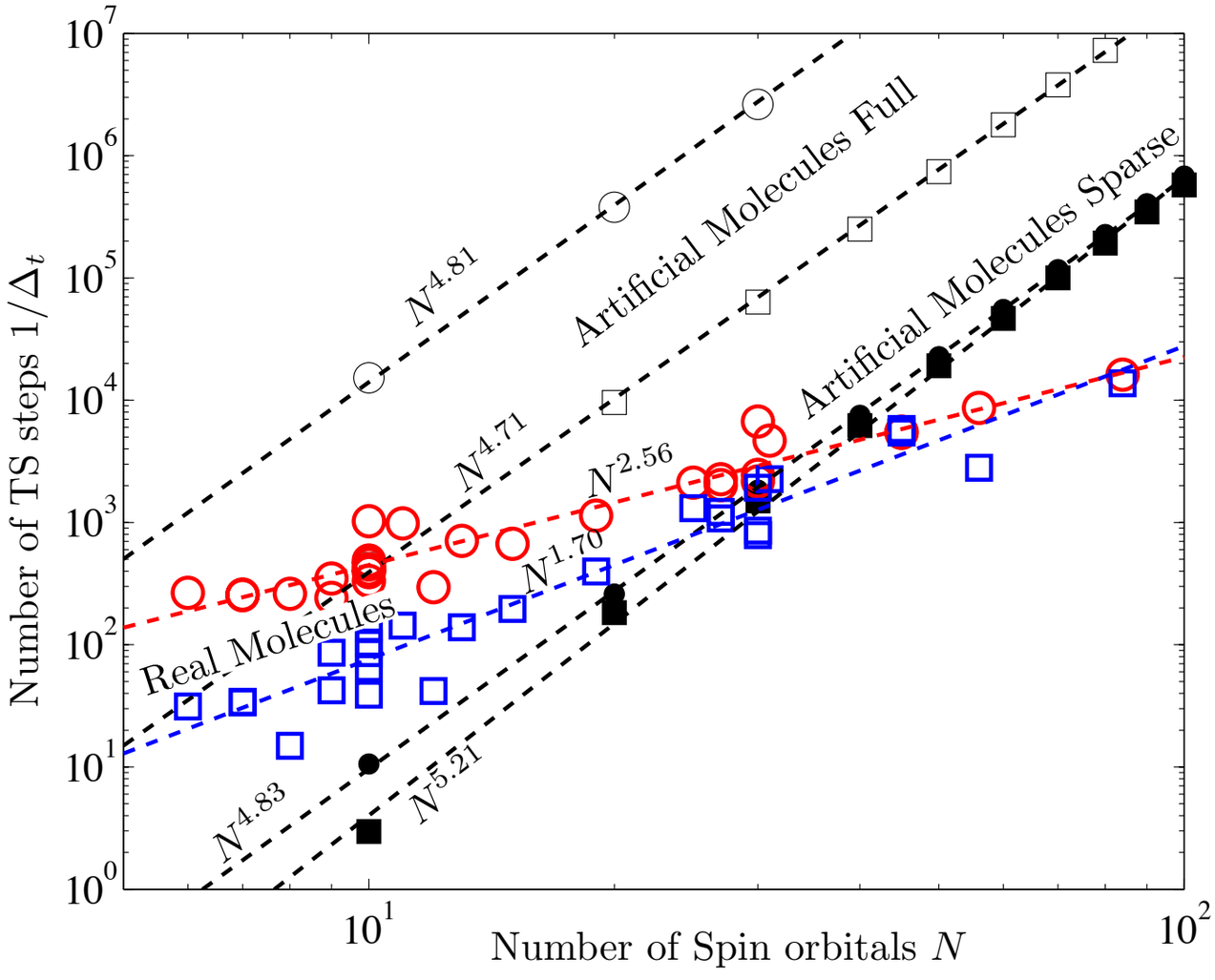
- **Improved phase estimation**
  - propagates by time *t/2*
  - no controlled rotations are needed
  - 2x fewer gates, 4x fewer rotations

$$\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)|\phi_n\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle e^{+iHt/2}|\phi_n\rangle+|1\rangle e^{-iHt/2}|\phi_n\rangle\right)=\frac{1}{\sqrt{2}}\left(e^{+i\phi/2}|0\rangle+e^{-i\phi/2}|1\rangle\right)|\phi_n\rangle$$

Optimization4: Reducing Trotter-Suzuki errors

D. Poulin *et* al., QIC (2015)

- Reorder the terms in the Trotter-Suzuki decomposition
- Multi-resolution Trotter scheme: longer time steps fort smaller terms
- Estimate tighter error bounds



Improvements to the quantum algorithms:

- Estimates for an example molecule: Fe_2S_2 with 118 spin-orbitals

Gate count	10^{18}
Parallel circuit depth	10^{17}
Run time @ 10ns gate time	30 years

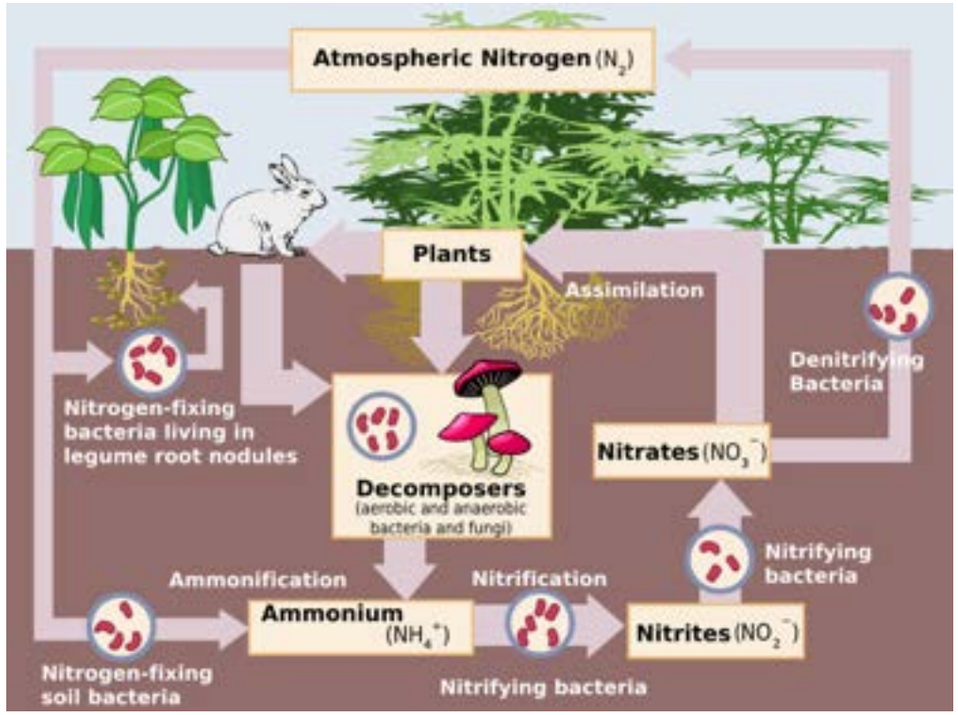
New gate count	10^{11}
Parallel circuit depth	10^{10}
Run time @ 10ns gate time	2 minutes

- Attempting to reduce the horrendous runtime estimates we achieved
  Wecker *et al.*, PRA (2014), Hastings *et al.*, QIC (2015), Poulin *et al.*, QIC (2015)

Cancelling of Jordan-Wigner strings:	O(*N*) reduction in gates
Nesting of terms:	O(*N*) reduction in circuit depth
Optimizing circuits:	4x reduction in gates
Smart interleaving of terms:	10x reduction in Trotter steps
Multi-resolution Trotter:	10x reduction in gates
Better quantum phase estimation:	4x reduction in rotation gates

Nitrogen fixation: a potential killer-app

- Ultimate problem:
  - Find catalyst to convert nitrogen to ammonia at room temperature
  - Reduce energy for conversion of air to fertilizer



- Current solution: Haber process (1909)
  - Requires high pressures and temperatures
  - Cost: 3-5% of the worlds natural gas production (1-2% of the world's annual energy)

- Quantum solution:
  - ~ 200 -400 qubits
  - Design a catalyst to enable inexpensive fertilizer production

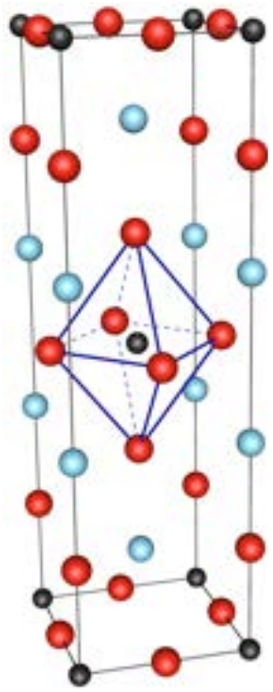
What about a room temperature superconductor?

	Full material
Bands per unit cell	≈ 50
Unit cells needed	20x20
Number of spin-orbitals	$N \approx 80'000$
Number of interaction terms	N^4
Cirucit depth scaling	$O(N^{5.5})$
Estimated runtime @ 10ns gate time	age of the universe

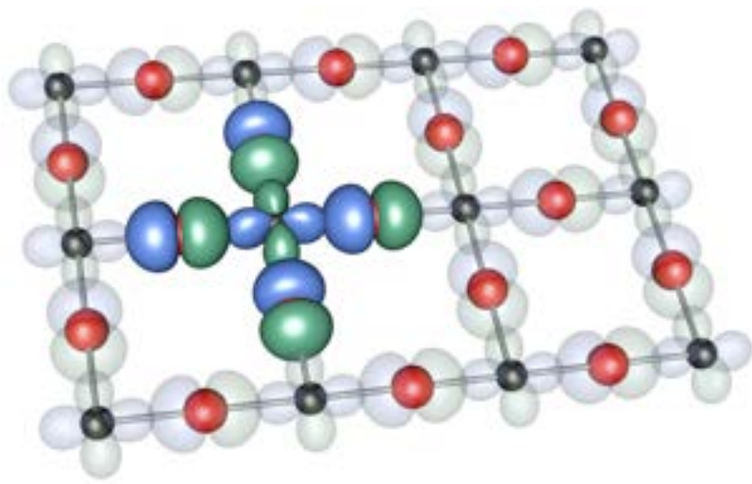
Reduce it to an effective model

Simplify to an effective model capturing the relevant physics

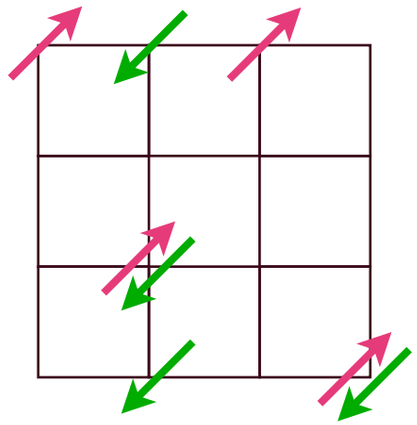
3D crystal structure



single 2D layer



simplified effective model

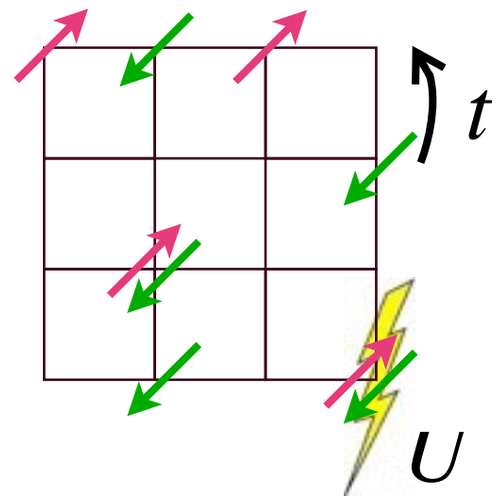


Can we efficiently solve such effective models on a quantum computer?

The Hubbard model

- A simple effective model with only O(*N*) terms

$$H = -t \sum_{\langle i,j \rangle, \sigma} (c^\dagger_{i,\sigma} c_{j,\sigma} + c^\dagger_{j,\sigma} c_{i,\sigma}) + U \sum_i n_{i,\uparrow} n_{i,\downarrow}$$

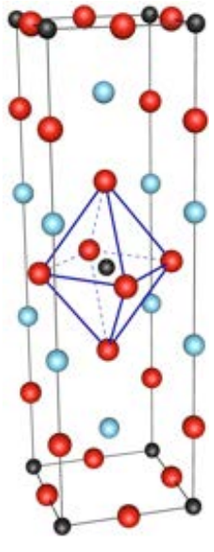


- Advantages
  - O(*N*) terms instead of O(N^4) terms
  - Can apply all terms in parallel => O(log *N*) circuit depth for one step
  - Much smaller energy range => shorter phase estimation times
  - Total scaling for adiabatic state preparation about O(*N*)

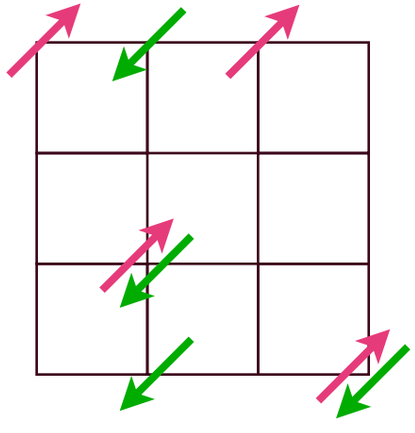
- Preparation and measurement need to be optimized
  - gaps to excitations
  - superconducting pair correlation functions (N^4 terms)
  - dynamic correlation functions

What about a room temperature superconductor?

	Full material	Hubbard model
Bands per unit cell	≈ 50	1
Unit cells needed	20x20	20x20
Number of spin-orbitals	$N \approx 80'000$	$N \approx 800$
Number of interaction terms	N^4	$O(N)$
Cirucit depth scaling	$O(N^{5.5})$	$O(1)$
Estimated runtime @ 10ns gate time	age of the universe	seconds



3D crystal structure



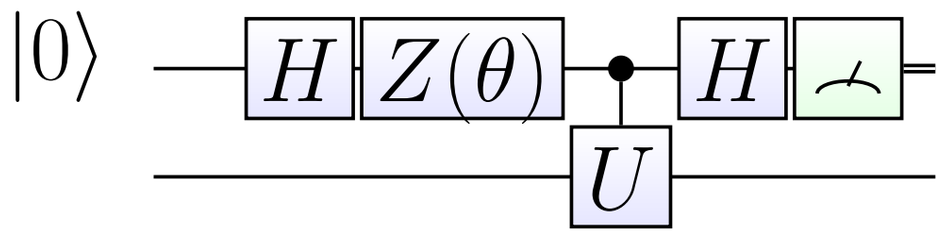
Hubbard model

Previous work on the Hubbard model

- Abrams and Lloyd, PRL (1997) suggested to use quantum computers for the time evolution under the Hubbard Hamiltonian
- Ortiz *et al*, PRA (2001) provided details on how to map fermions to qubits and how to measure some observables
- Open questions after these seminal papers:
  - preparing good trial wave functions
  - mapping the time evolution onto actual quantum gates
  - estimating the circuit depth (gate count)

Direct measurements

- After preparing the ground state we can measure any desired quantity *U* which we can implement as an efficient circuit



- However, every measurement only gives a single bit!

$$\Delta O = \sqrt{\frac{\mathrm{Var} O}{M}} \Rightarrow M = O\left(\frac{1}{\epsilon^2}\right)$$ measurements are needed

- Solution: perform a quantum amplitude estimation to get the value (Knill, Somma, Ortiz, PRA (2007))

$$t = O\left(\frac{1}{\epsilon}\right)$$ measurement time is needed

- This still destroys the state and we have to (expensively) re-prepare it

Nondestructive measurements (approach I)

Wecker *et al*, PRA (2015)

- Make use of the fact that the ground state is an eigenstate of the Hamiltonian to perform non-destructive measurements

- Measurements using Hellman-Feynman theorem

$$\frac{dE_{GS}(\lambda)}{d\lambda} = \left\langle \Psi_{GS}(\lambda) \right| \frac{dH(\lambda)}{d\lambda} \left| \Psi_{GS}(\lambda) \right\rangle$$

- Add the observable to be measured as a perturbation $H(\lambda) = H + \lambda O$

- Adiabatically evolve the wave function $\left|\Psi_{GS}(0)\right\rangle \rightarrow \left|\Psi_{GS}(\epsilon)\right\rangle \rightarrow \left|\Psi_{GS}(-\epsilon)\right\rangle$

- Measure the energy and calculate the difference $\left\langle \Psi_{GS} \right| O \left| \Psi_{GS} \right\rangle \approx \frac{E_{GS}(\epsilon) - E_{GS}(-\epsilon)}{2\epsilon}$

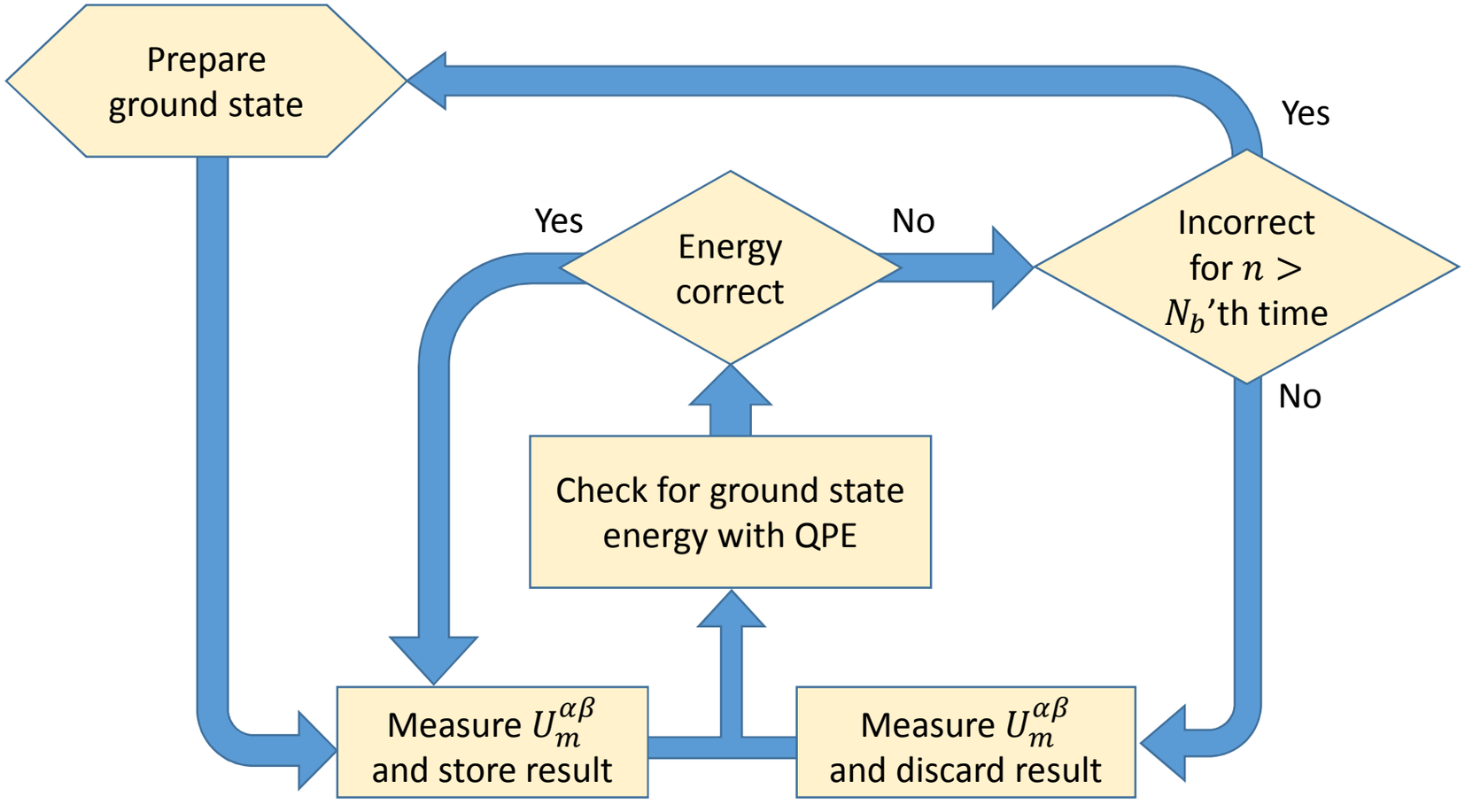
- Non-destructive and only $O\left(\frac{\log \epsilon}{\epsilon}\right)$ measurements are needed

Nondestructive measurements (approach II)

Wecker *et al*, PRA (2015)

- Recover the ground state (GS) after a destructive measurement

Efficient if we measure only a single bit
requires time $O(1/\epsilon)$ if performed coherently



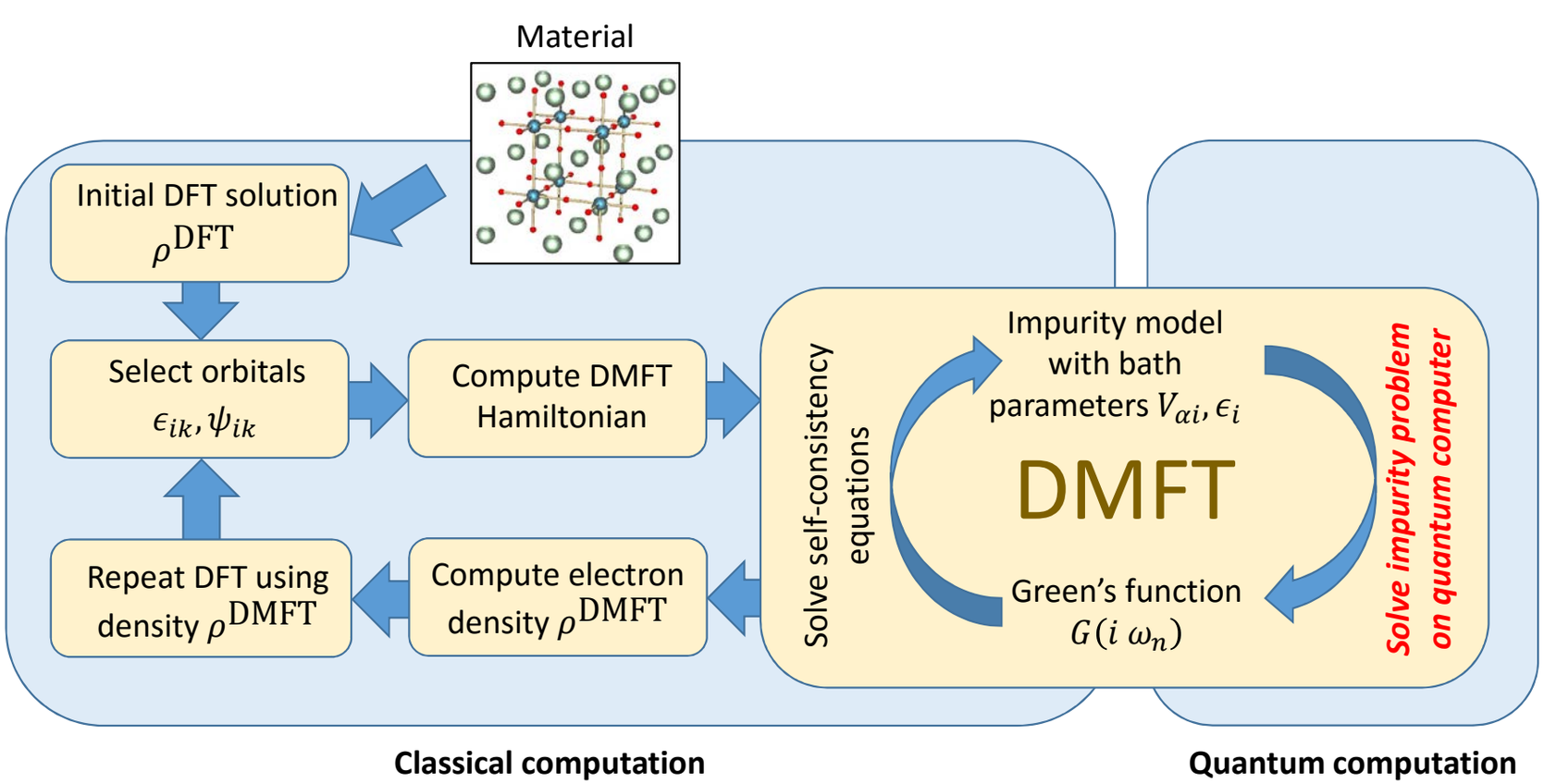
From the Hubbard model to materials

	Full material	Hubbard model
Bands per unit cell	≈ 50	1
Unit cells needed	20x20	20x20
Number of spin-orbitals	$N \approx 80'000$	$N \approx 800$
Number of interaction terms	N^4	$O(N)$
Cirucit depth scaling	$O(N^{5.5})$	$O(1)$
Estimated runtime @ 10ns gate time	age of the universe	seconds

- The Hubbard model teaches us fundamental mechanisms but not the properties of a real material.
- We gain valuable insights but not quantitative predictive power
- Full *ab-initio* simulation of a correlated material is too complex even on quantum hardware

Solution: a hybrid approach to materials simulation

- Use classical algorithms on peta-scale hardware to derive realistic effective realistic models of the important bands in a material
- Use quantum algorithms to solve these models on a quantum computer
- Iterate by using the results to improve the effective models



And develop it into a killer-app

7. Run the quantum algorithm to solve the problem

6. Embed into specific hardware and check runtime

5. Add error correction and check overhead and resources

4. Optimize code until logical circuit depth < 10^{14}

3. Check for quantum speedup

2. Implement all oracles and subroutines

1. Find quantum algorithm with quantum speedup

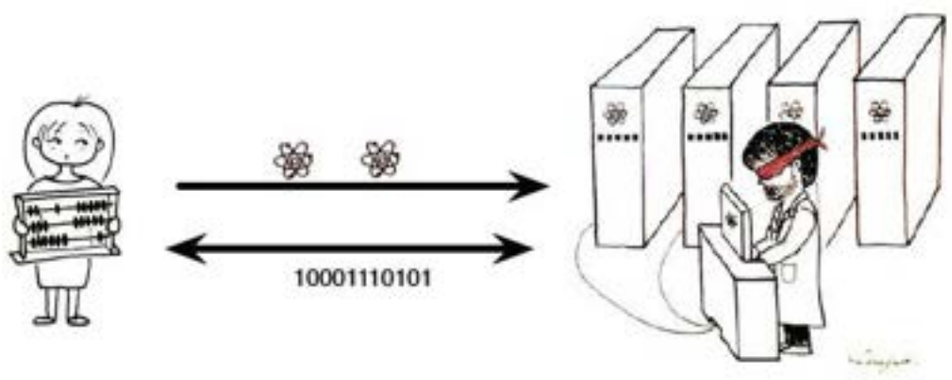
Quantum computing as post-exa-scale technology

- Identifying killer-apps for quantum computing is challenging
  - the problem has to be hard enough that it cannot be solved on an exascale machine
  - the problem has to be amenable to quantum acceleration
  - the crossover scale has to be short enough to make it useful

- Potential applications
  - factoring and code breaking (relatively easy but limited use)
  - quantum lattice models (straightforward!)
  - quantum chemistry and material simulations (challenging but enormous potential)
  - solving linear systems (can't we solve them well enough already?)
  - others??? machine learning???

- It is imperative to do realistic resource estimates
- Asymptotic scaling is not enough!
- It is time to view quantum algorithms from an applications perspective!

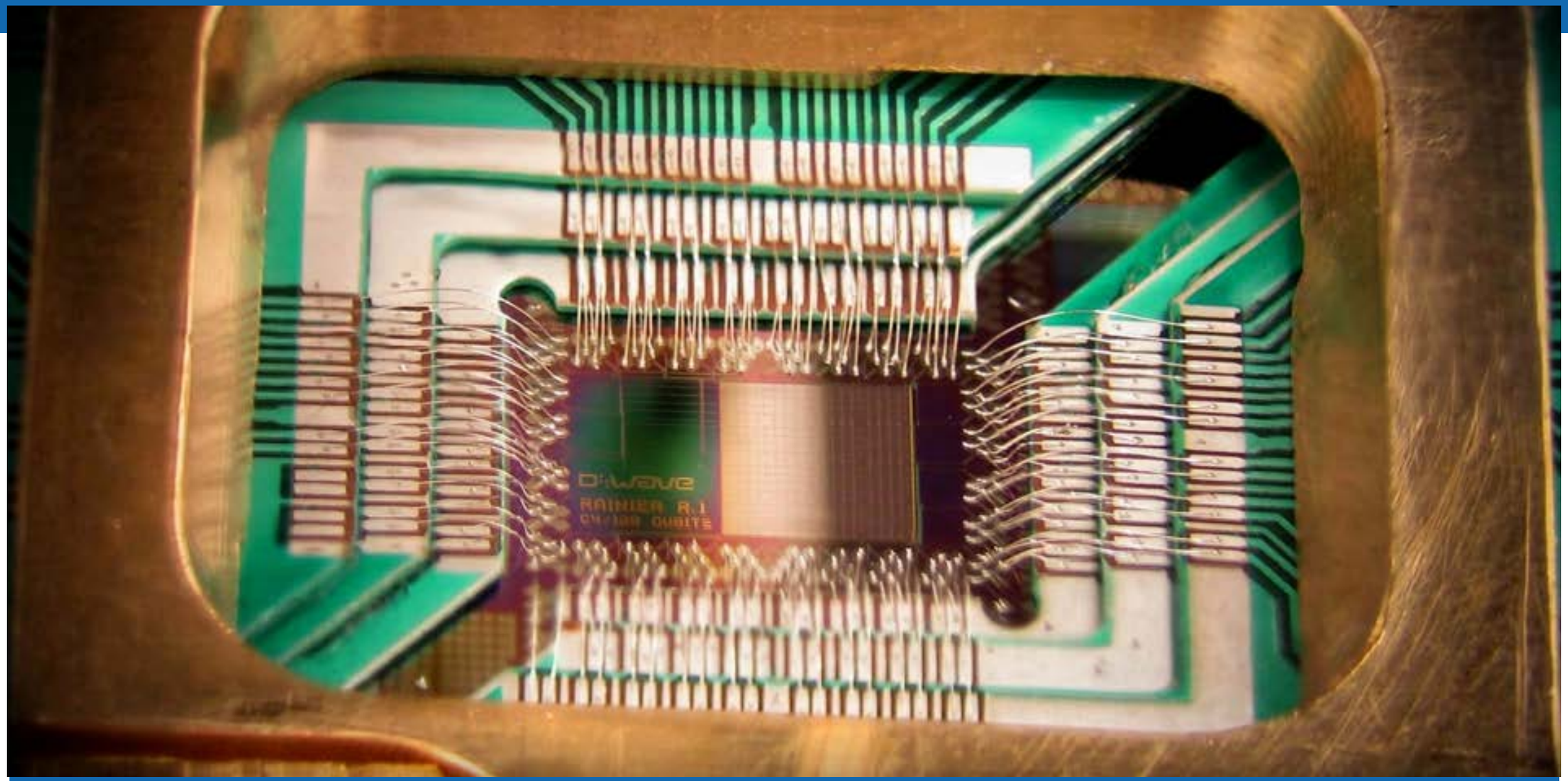
There is much more!



Blind quantum computing and search
(Broadbent, Fitzsimons, Kashefi)

Quantum money
(Aaronson, Farhi *et al*)

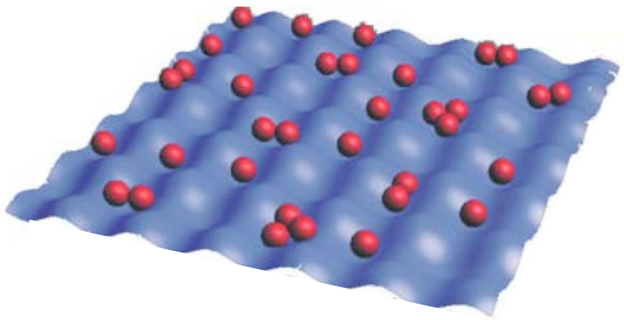
Cloud providers **cannot** know what the user does



Quantum annealing

Matthias Troyer

Analog and digital quantum devices



Quantum simulators

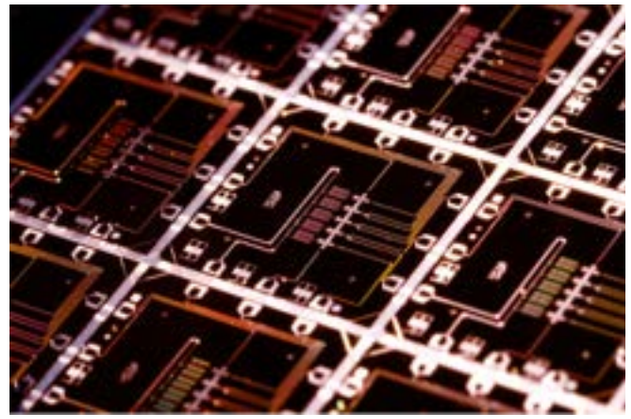
solve quantum models

Quantum annealer

solve hard optimization problems?

Quantum random numbers

perfect randomness



Quantum computers?

solve quantum models (R. Feynman)

factor integers (P. Shor)

...

Quantum encryption

secure communication

FEBRUARY 17, 2014

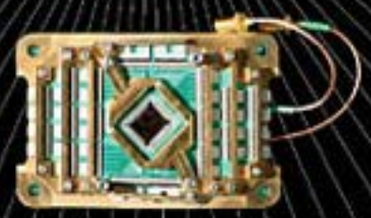
French Advances / My Doctor Fired Me / Love App-tually

TIME

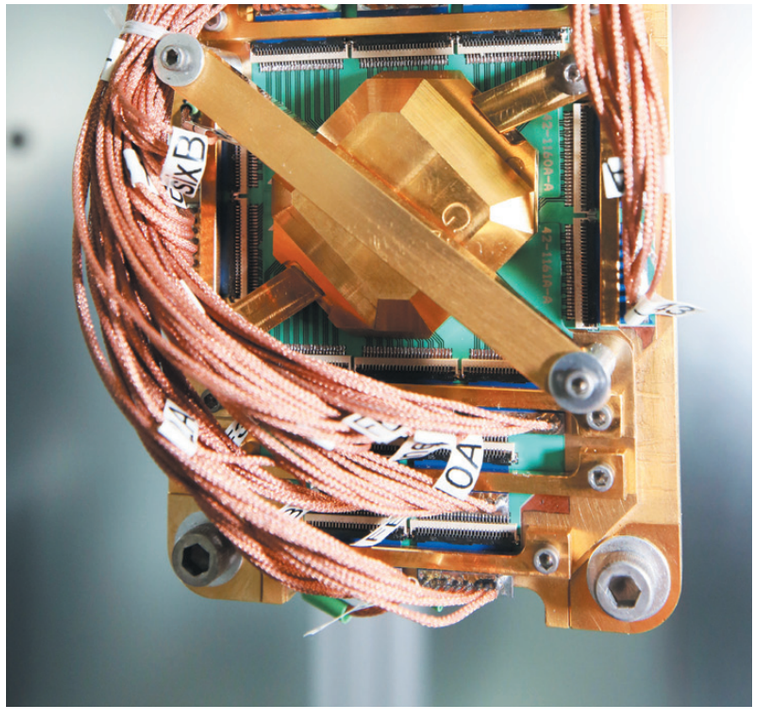
IT PROMISES TO SOLVE SOME OF HUMANITY'S MOST COMPLEX PROBLEMS. IT'S BACKED BY JEFF BEZOS, NASA AND THE CIA. EACH ONE COSTS $10,000,000 AND OPERATES AT 459° BELOW ZERO. AND NOBODY KNOWS HOW IT ACTUALLY WORKS

THE INFINITY MACHINE

BY LEV GROSSMAN



THE QUANTUM COMPANY



3,600 times faster than classical computers at some tasks

COURTESY OF D-WAVE SYSTEMS INC.

Quantum annealer. To solve a problem, D-Wave's chip seeks the lowest energy state of 512 interacting quantum bits, or qubits, fashioned from tiny rings of superconductor.

Quantum or not, controversial computer runs no faster than a normal one

It’s in a quantum superposition!

Quantum computing

Faster, slower—or both at once?

The first real-world contests between quantum computers and standard ones

CHIPMAKERS dislike quantum mechanics. Half a century of Moore’s law means their products have shrunk to the point where they are subject to the famous weirdness of

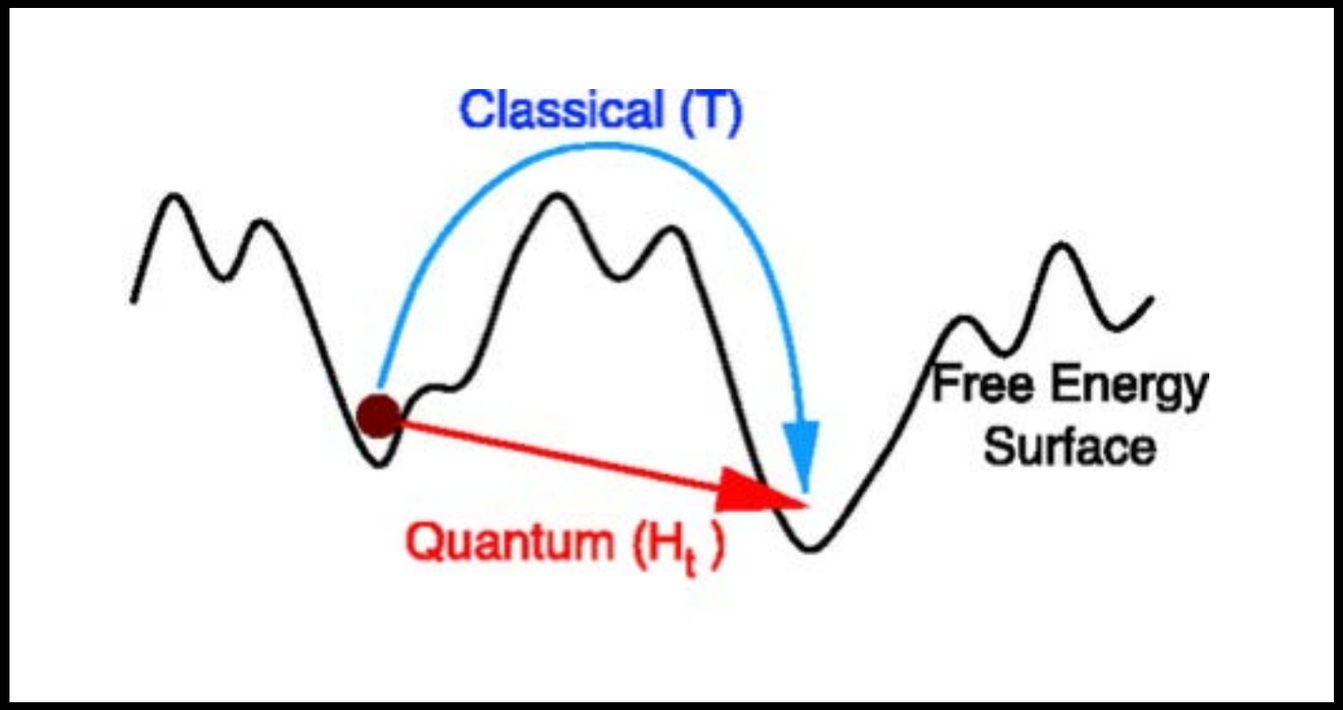
Daily Mail

Google tries to save the world: Internet giant explains how its move into quantum computing could solve global warming

Google hopes it will help develop sophisticated artificial life, and find aliens

D-Wave computers run on a 512-qubit processor.
As a comparison, PCs found in homes run on 32-bit or 64-bit processors.

Quantum annealing of Ising spin glasses



A device to solve Ising spin glass problems

Find the configurations which minimize the energy

$$H = \sum_{ij} J_{ij} s_i s_j + \sum_i h_i s_i \qquad \text{with} \qquad s_i = \pm 1$$

This problem is (nondeterministic polynomially) NP-hard, meaning that many interesting hard problems can be mapped onto it

- traveling salesman problem
- portfolio optimization
- factorization of integers
- graph isomorphisms
- and many more …

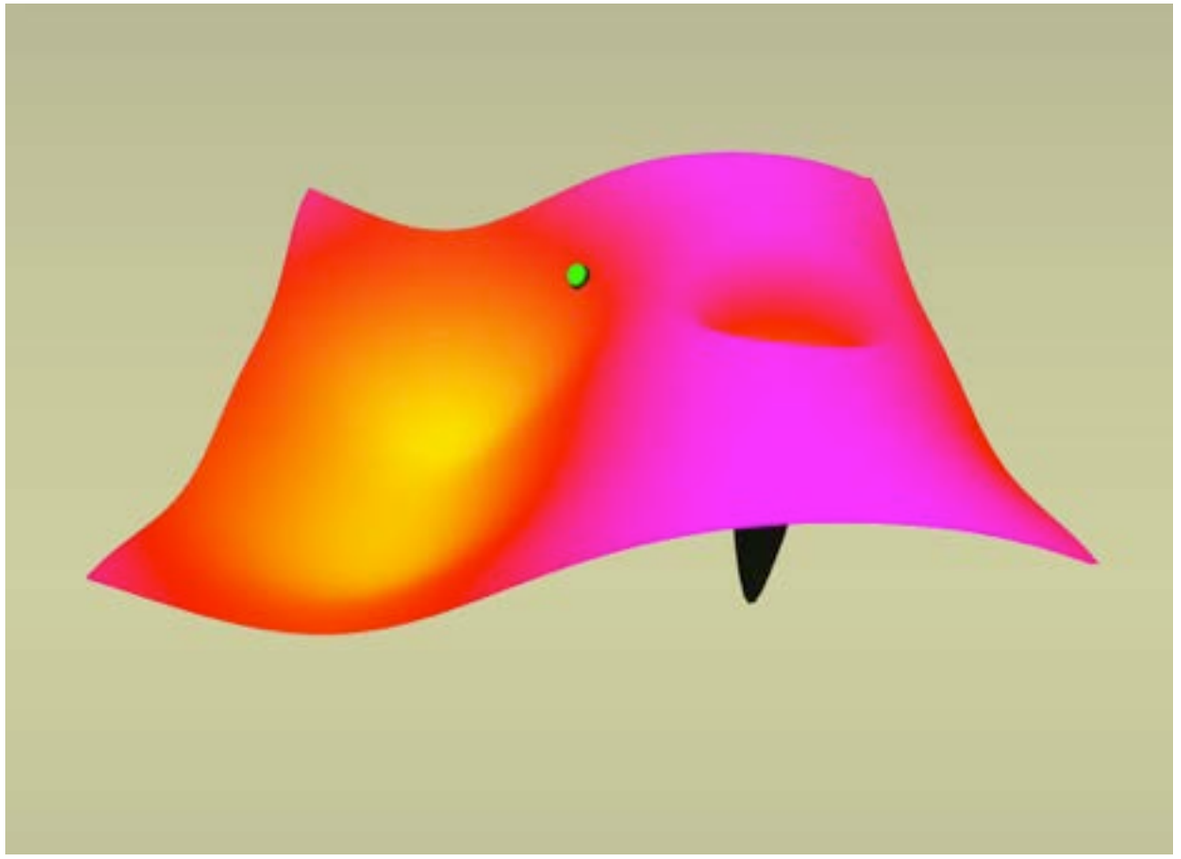
Can a quantum device solve these problems faster than a classical one?

Annealing and simulated annealing

Annealing

A 7000 year old neolithic technology

Slowly cool metal or glass
to improve its properties



Simulated annealing

Kirkpatrick, Gelatt and Vecchi, Science (1983)

A 30 year old optimization technique

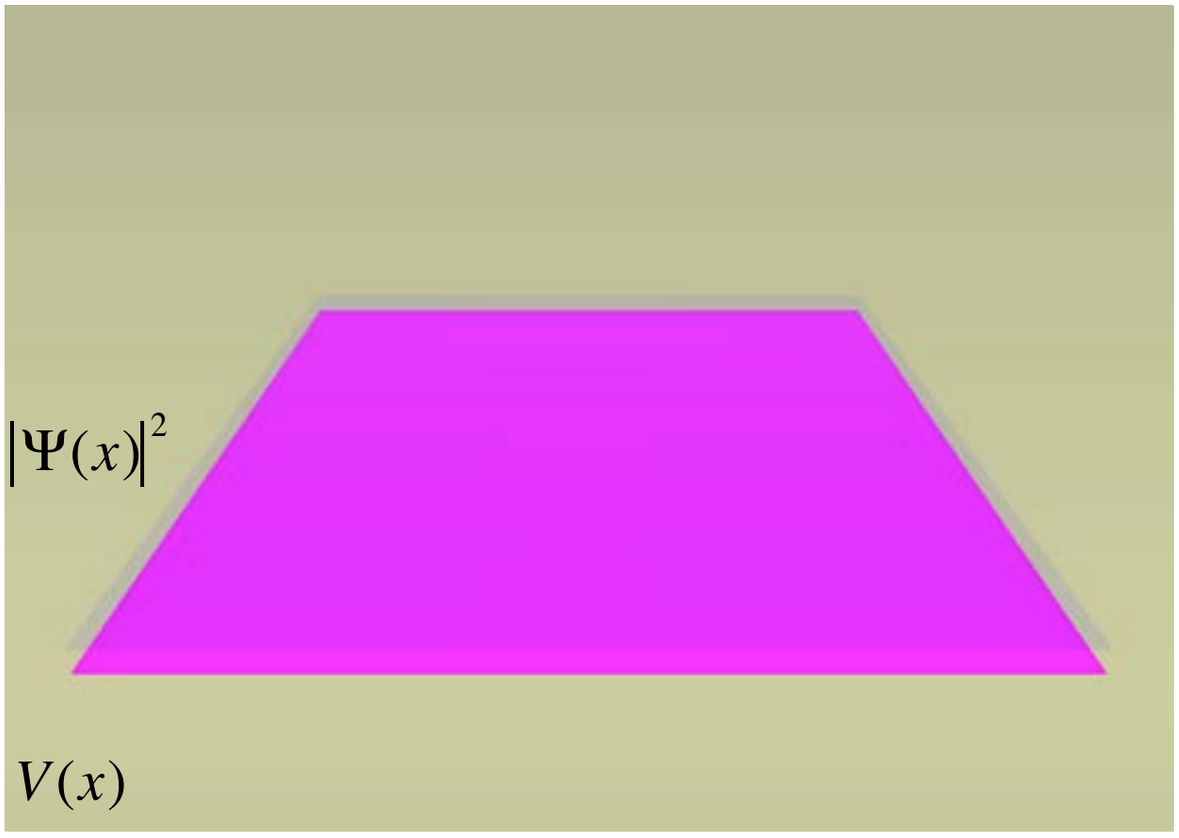
Slowly cool a model in a Monte Carlo simulation
to find the solution to an optimization problem

We don’t always find the global minimum and have to try many times

Quantum adiabatic algorithm

Farhi, Goldstone, Gutmann and Sipser (2000)

Quantum adiabatic theorem: if we slowly change the Hamilton a quantum system remains in the ground state of the instantaneous Hamiltonian



have to evolve sufficiently slowly!

Turning on the potential the wave function concentrates around the minima of the potential

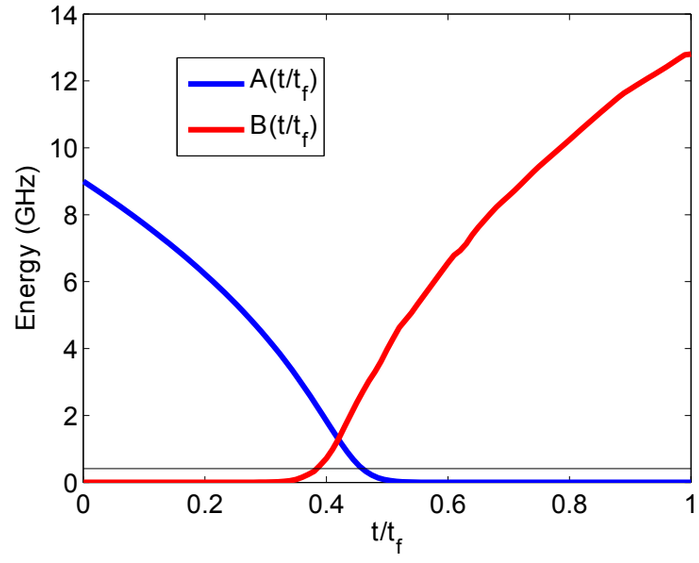
Quantum annealing for the Ising spin glass

Add a transverse magnetic field to induce quantum fluctuations

$$H(t) = B(t)\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z - A(t)\sum_i \sigma_i^x$$

Initial time t=0: all spins aligned with the transverse field

Final time $t=t_f$: ground state of the Ising spin glass



Ground state version: quantum adiabatic algorithm of *Farhi et al, (2000)*

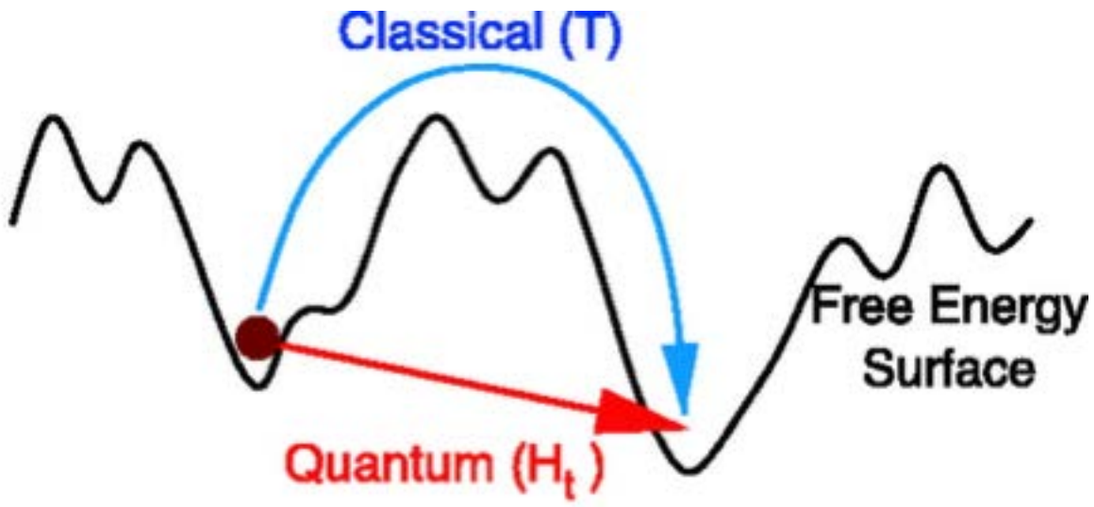
Finite temperature version: quantum annealing by *Brooke et al, (1999)*

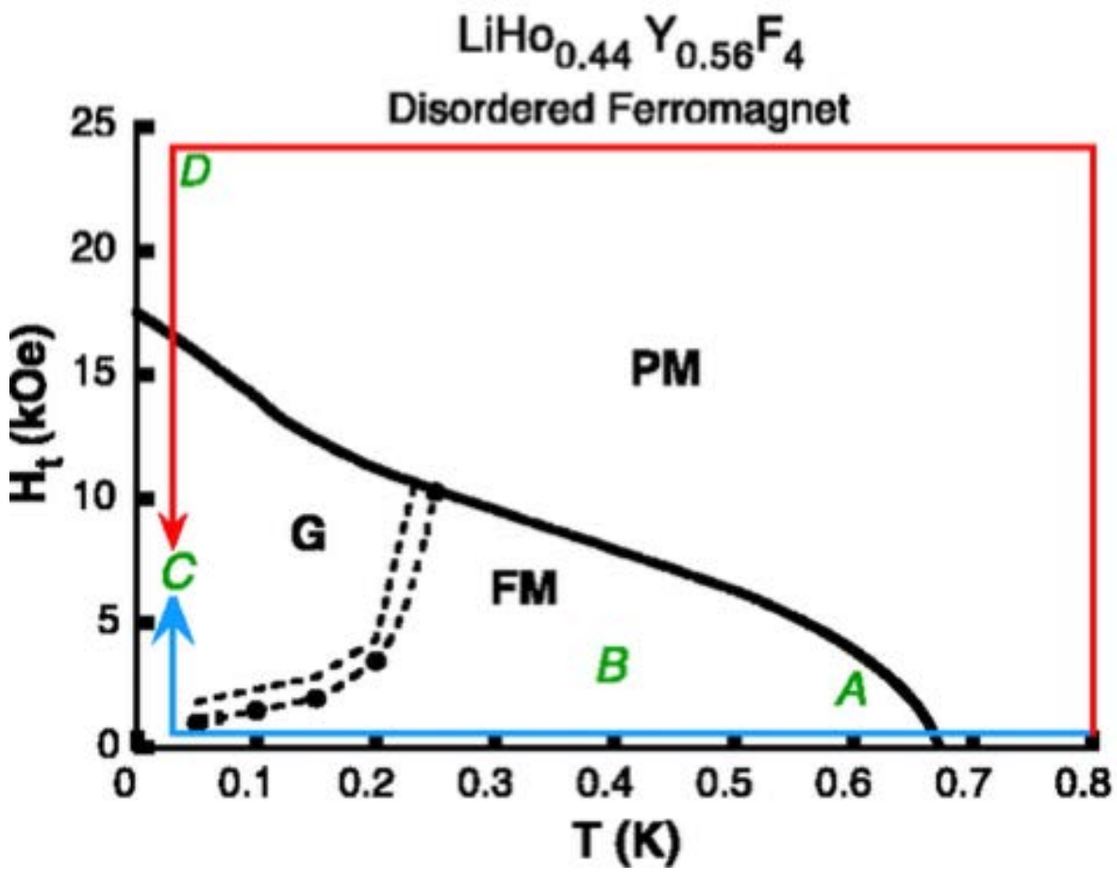
Simulated quantum annealing: same idea in a QMC simulation, *Santoro et al, (2002)*

Quantum annealing in experiments

Brooke, Bitko, Rosenbaum, Aeppli, Science (1999)

Anneal at very low temperatures by changing a *quantum* control parameter





Cool a spin glass by sweeping a magnetic field at low temperatures

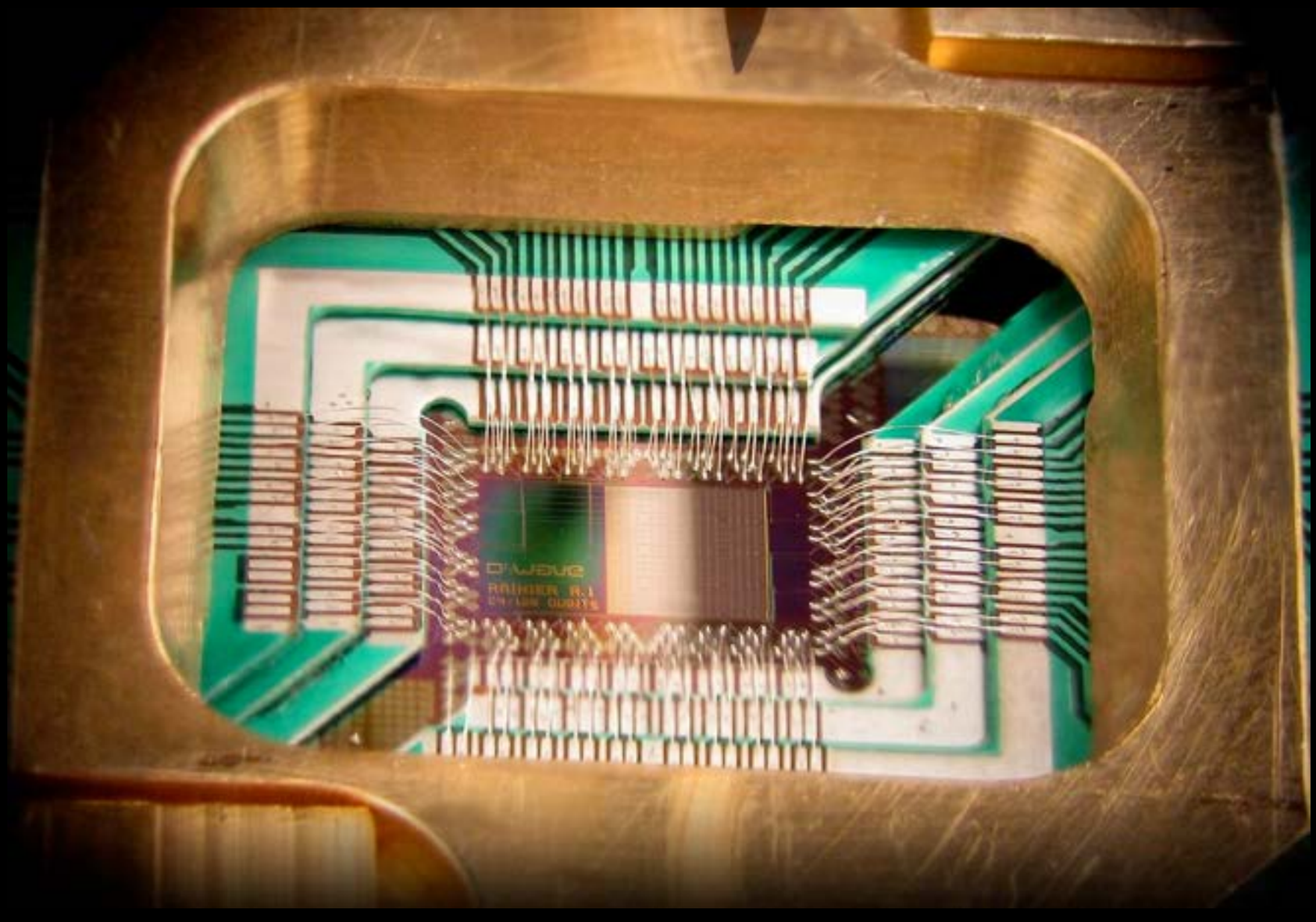
Three open questions about quantum annealing for typical (not worst case) problems

1. Does adiabatic quantum computing for the Ising spin glass have any speedup over classical algorithms?

2. Does finite-temperature quantum annealing have any speedup over classical algorithms?

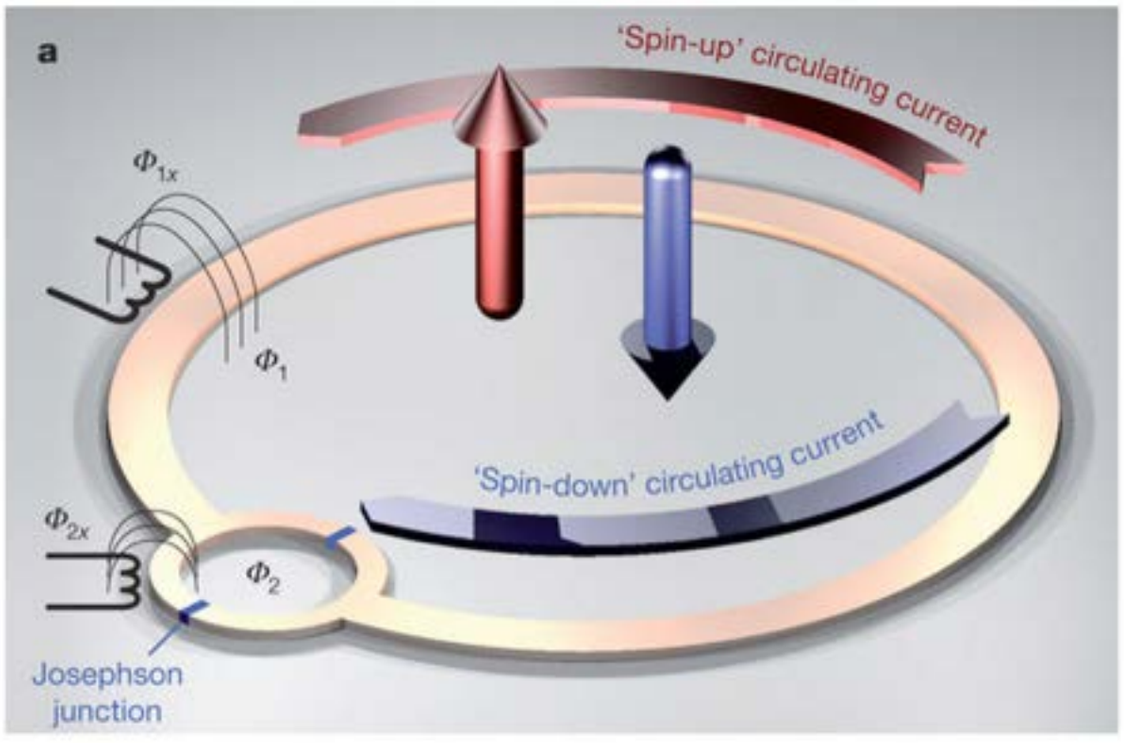
3. Does the implementation in the D-Wave devices have any speedup over classical algorithms?

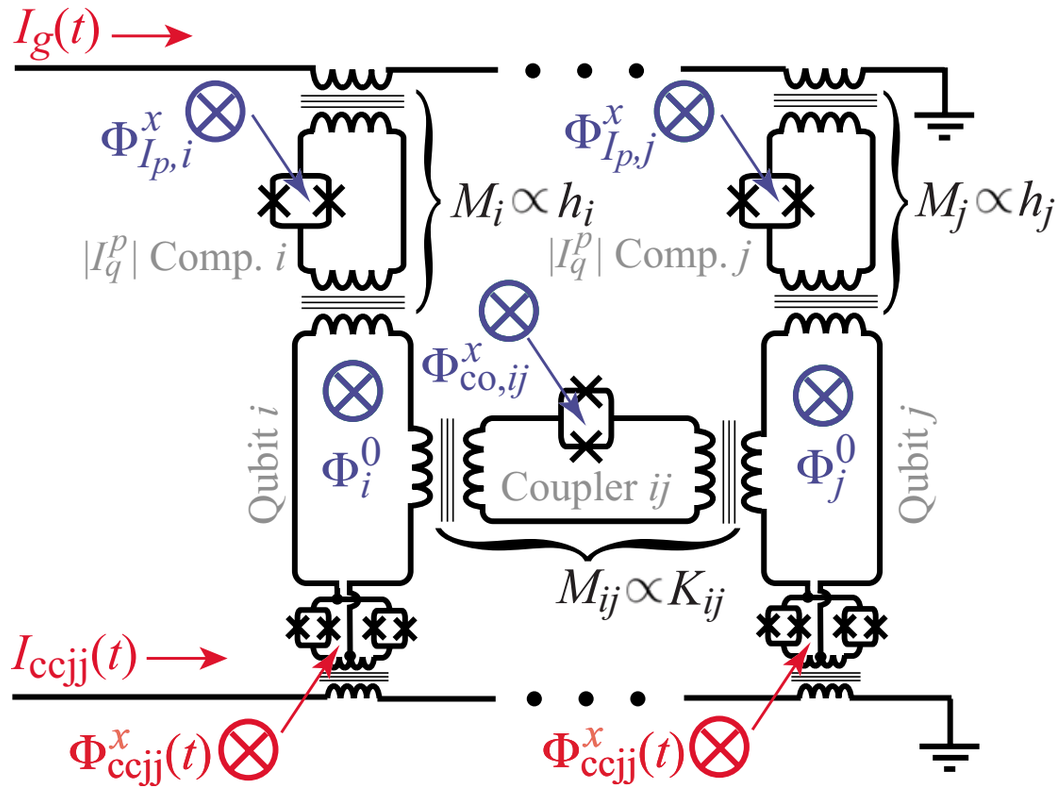
The D-Wave device



The D-Wave device: superconducting flux qubits

Implement Ising spins by flux quanta through superconducting rings
Programmable inductive couplings between flux qubits

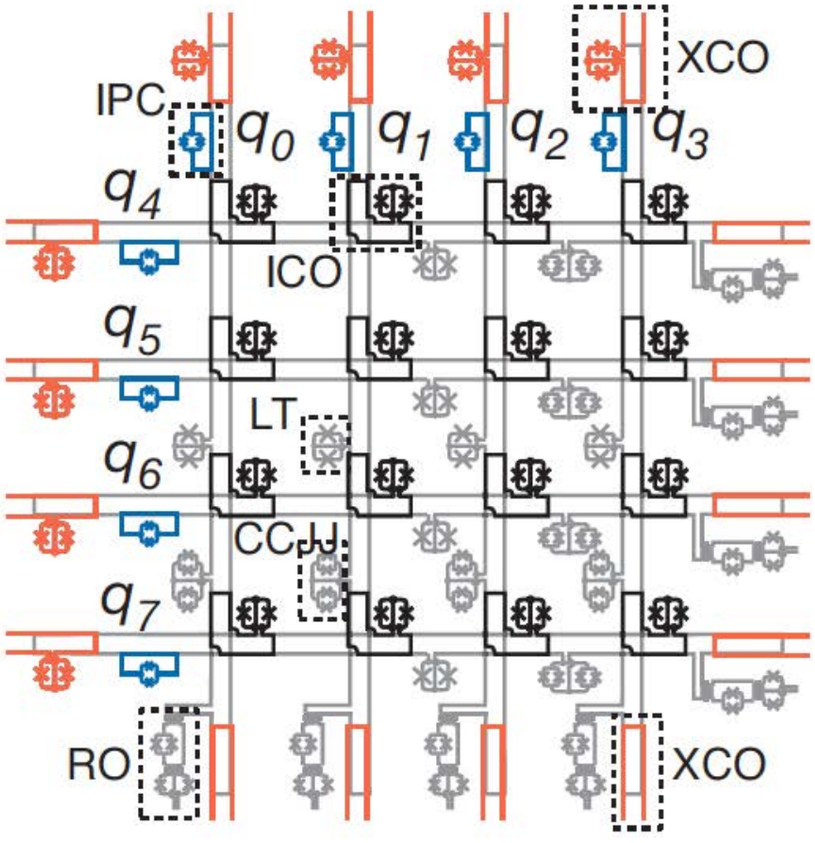


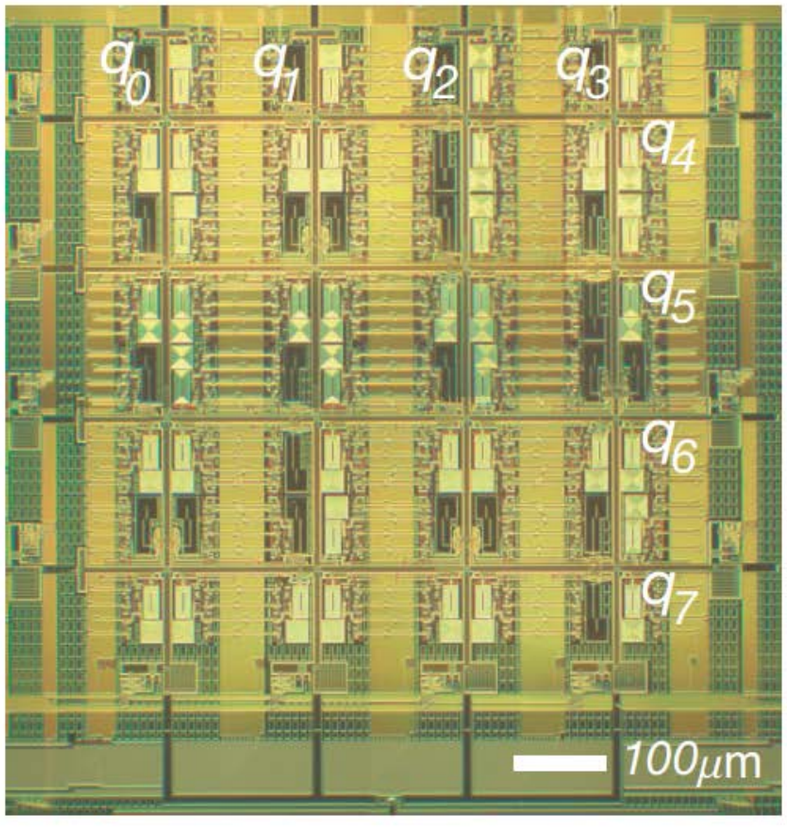


100’000 Josephson junctions
4’000 digital to analog converters

A unit cell of the device

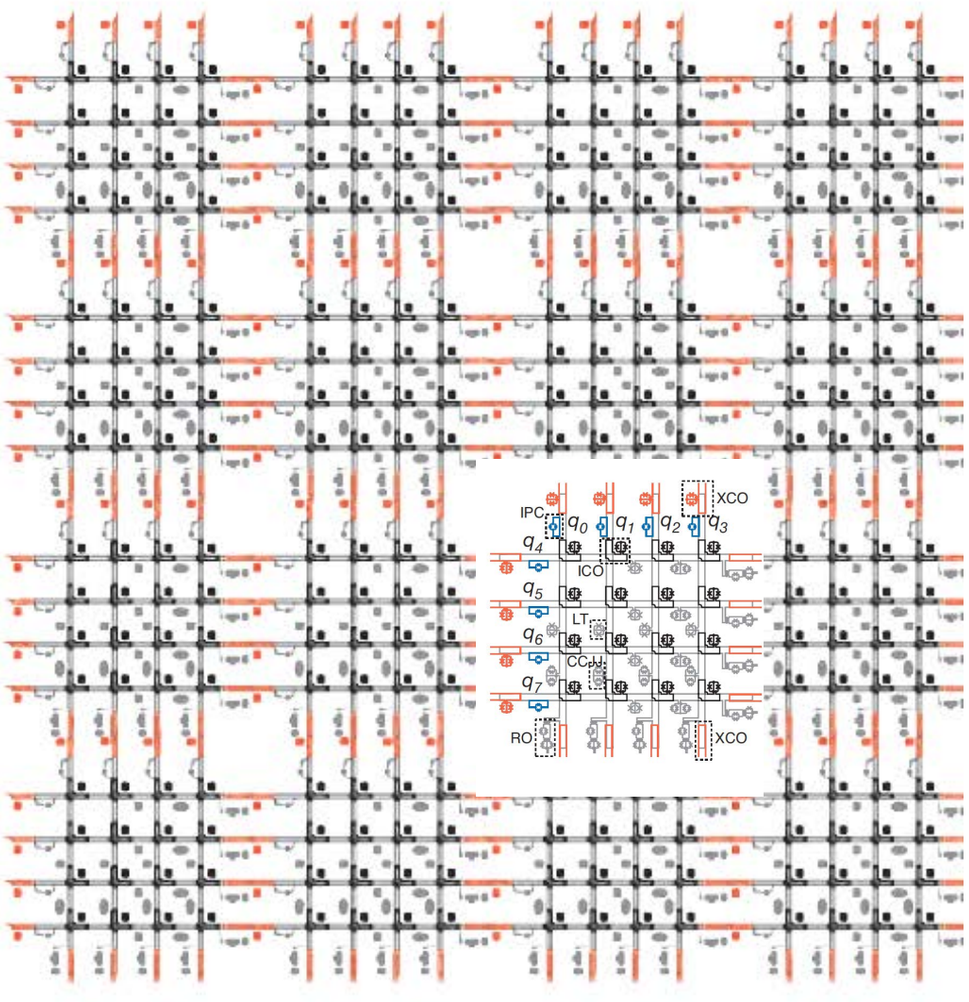
Arrange four qubits horizontally and four vertically
Obtain a bipartite fully connected graph of 8 spins

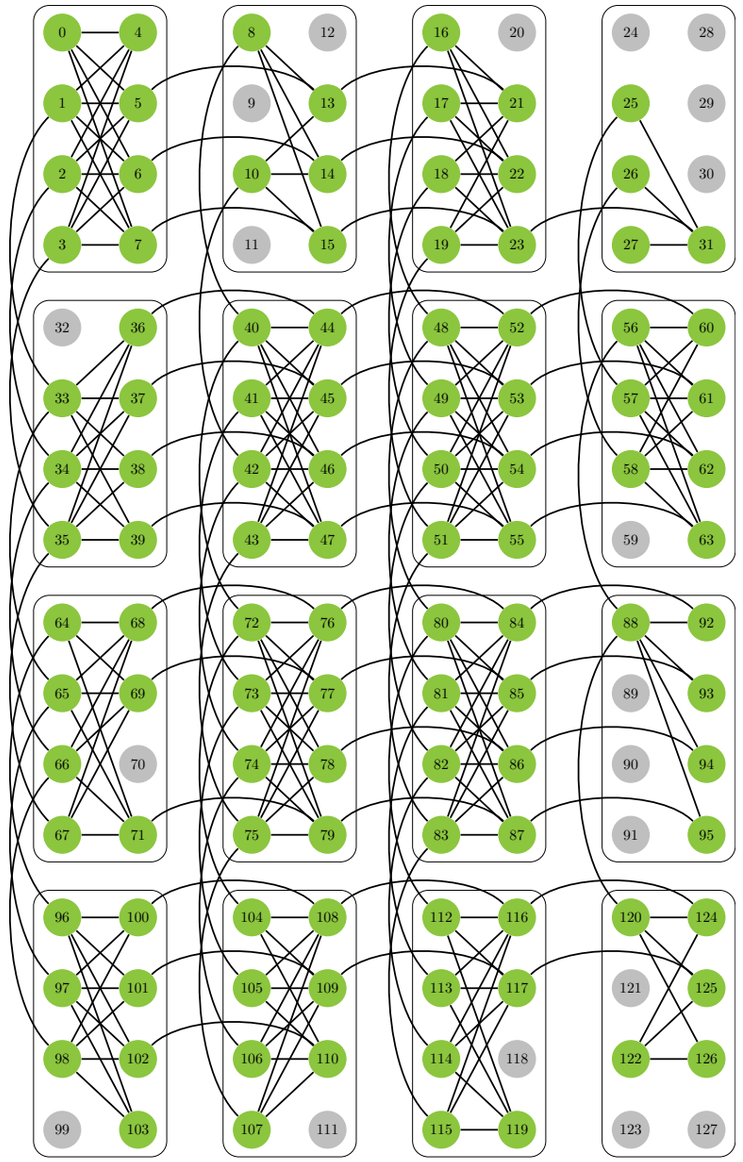




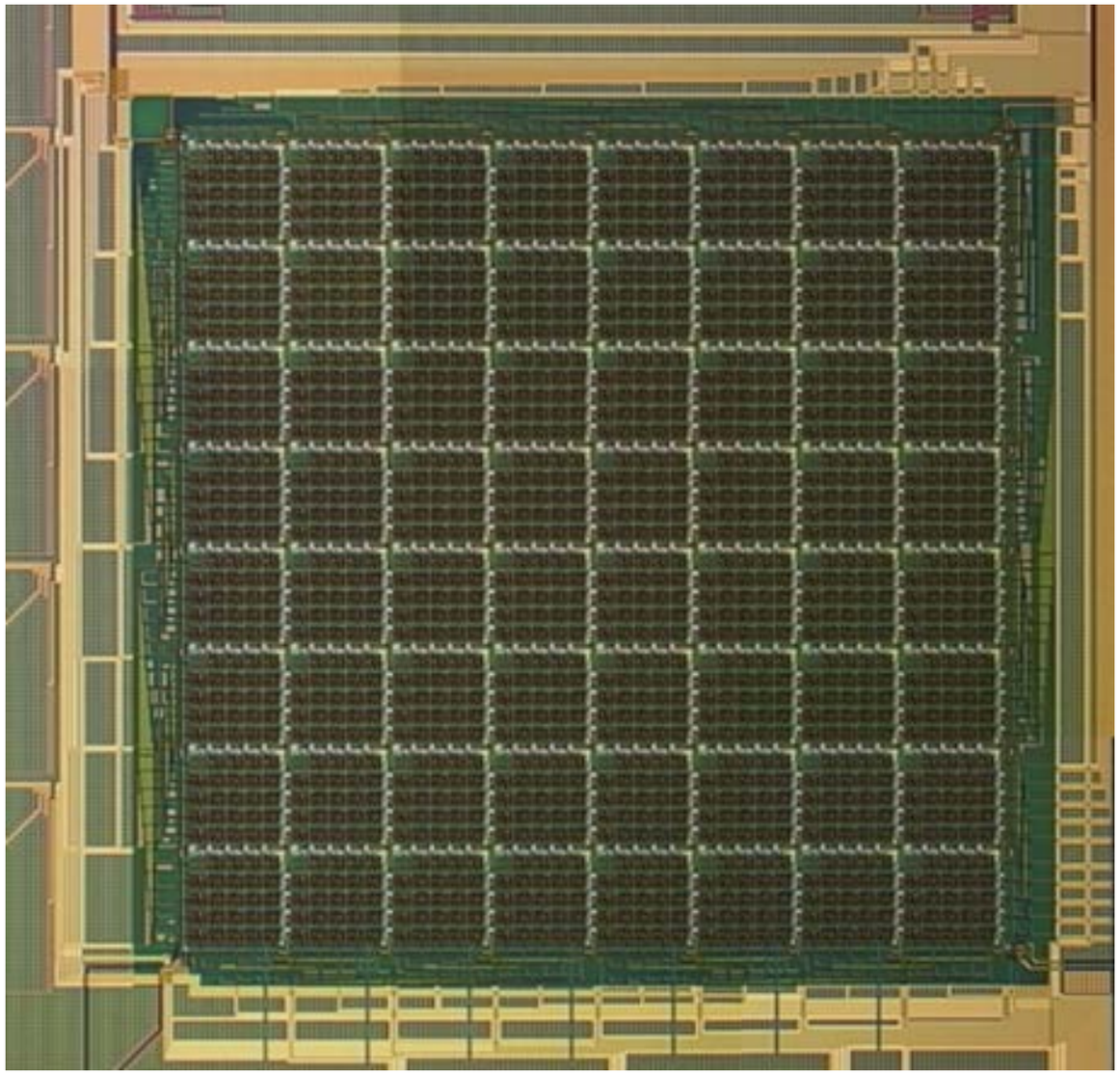
The full chimera graph: 4x4 unit cells = 128 qubits

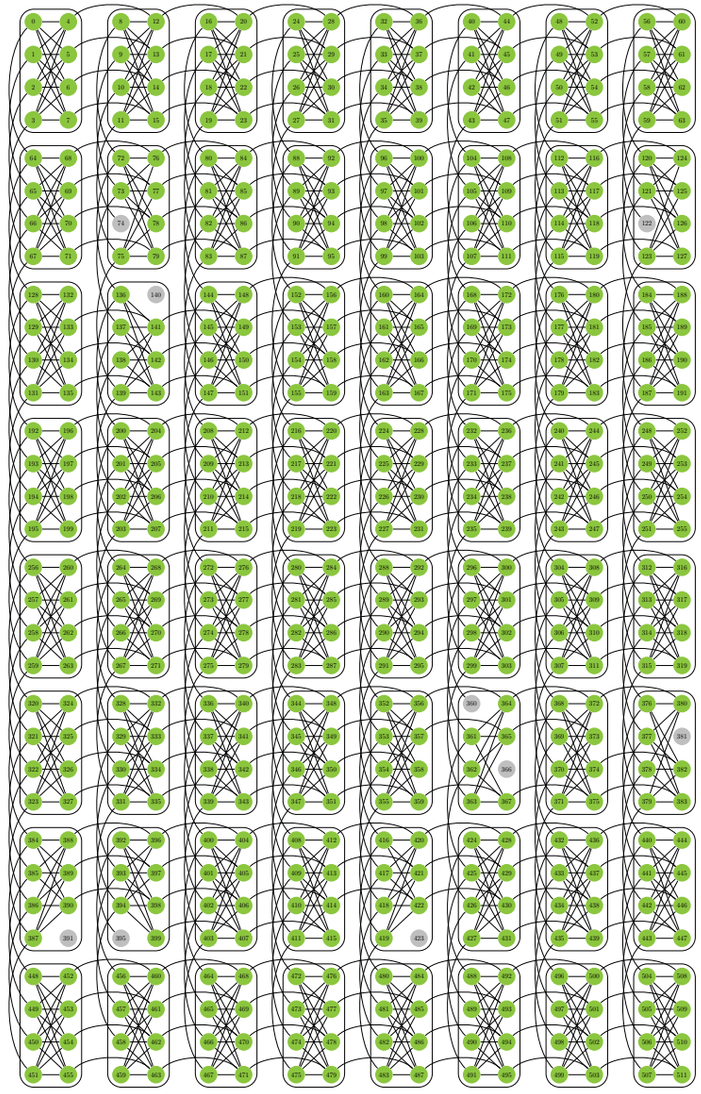
Couple horizontal qubits horizontally to the neighbors
Couple vertical qubits vertically to the neighbors
108 of 128 qubits worked



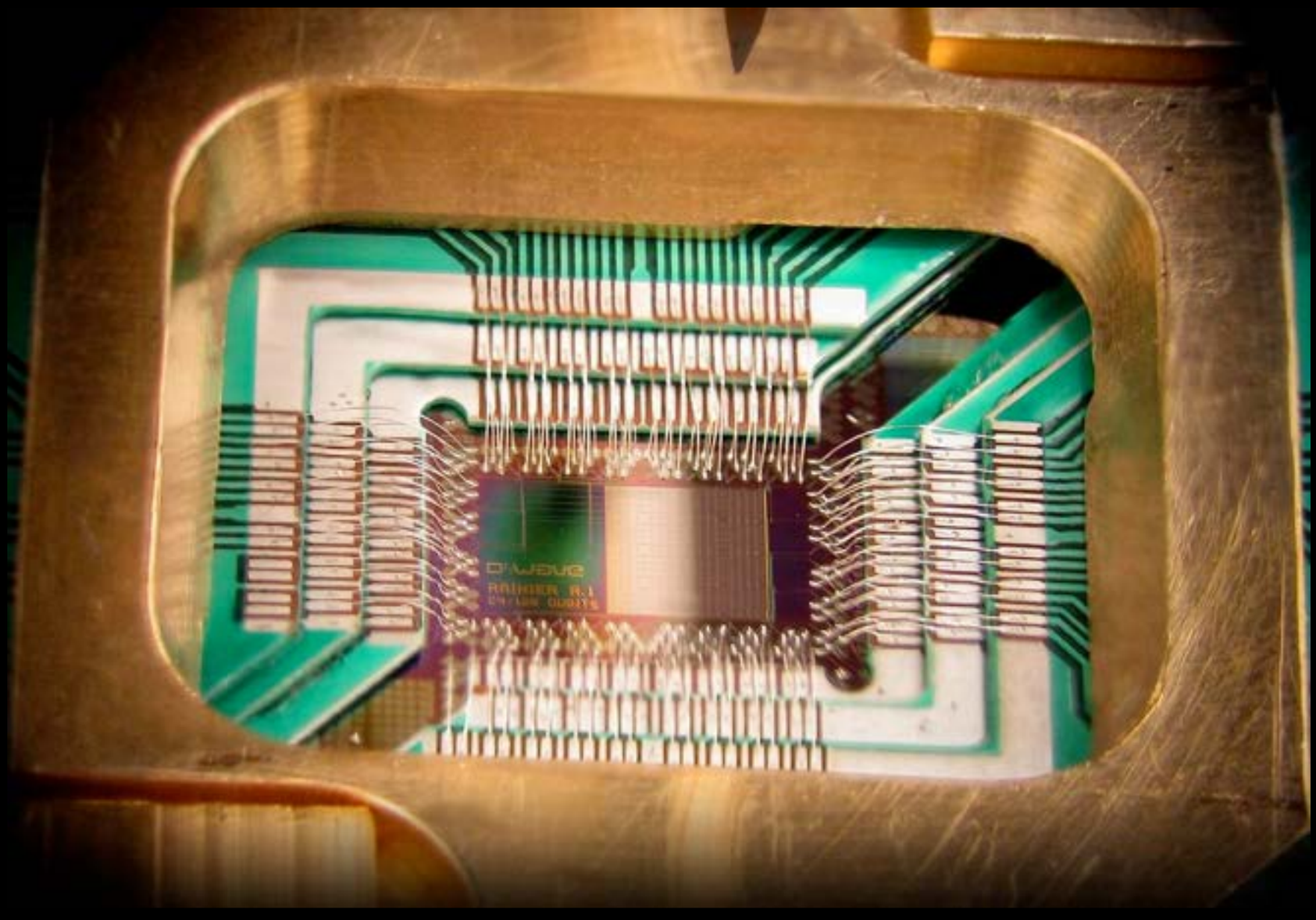


D-Wave Two: 8x8 unit cells with 512 qubits





What does it do?



Perform experiments to answer these questions

- **Hypothesis 1: D-Wave is a classical annealer**
  - Decoherence is so strong that quantum effects are irrelevant and it is just thermal.
  - Experimental test: compare to a simulated classical annealer

- **Hypothesis 2: D-Wave performs classical (mean field) spin dynamics**
  - The qubits are just classical spins precessing in the magnetic field.
  - Experimental test: compare to a simulated classical annealer

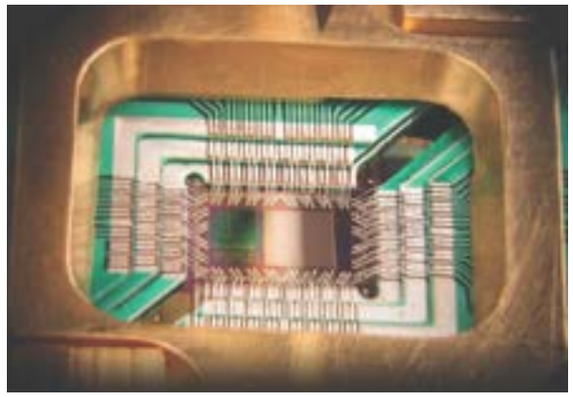
- **Hypothesis 3: D-Wave is an incoherent quantum annealer**
  - Decoherence is weak enough to allow a quantum model to be realized, but quantum coherence is unimportant for the annealing.
  - Experimental test: compare to a simulated quantum annealer

- **Hypothesis 4: D-Wave is a quantum annealer with quantum speedup**
  - Quantum coherence provides advantages in tunneling through barriers
  - Experimental test: scaling with problem size is better than that of a simulated annealer

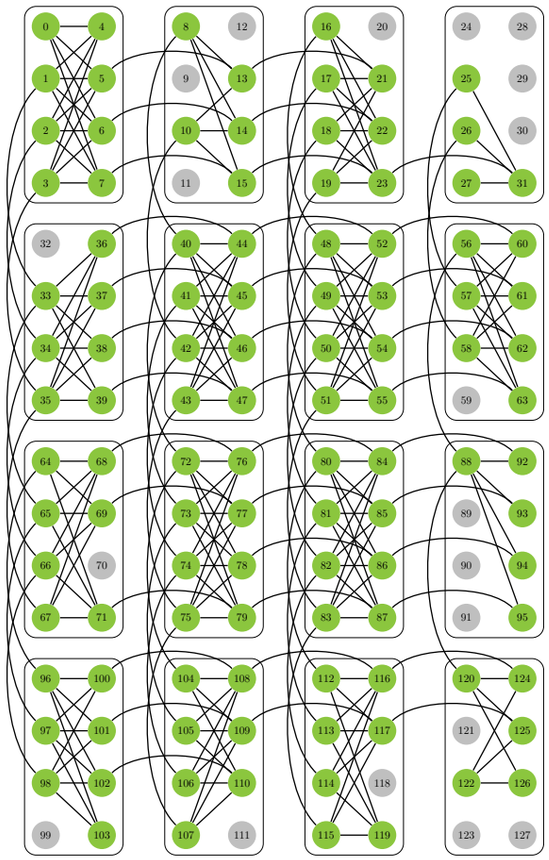
Our experiments

Find hard test problems for the machine to solve

random ±1 couplings on all bonds of the chimera graph



hundred million experiments
on D-Wave One



one billion simulations
classical and quantum Monte Carlo

1000 choices of couplings for each problem size
1000 repetitions of the annealing
vary the annealing time and schedule

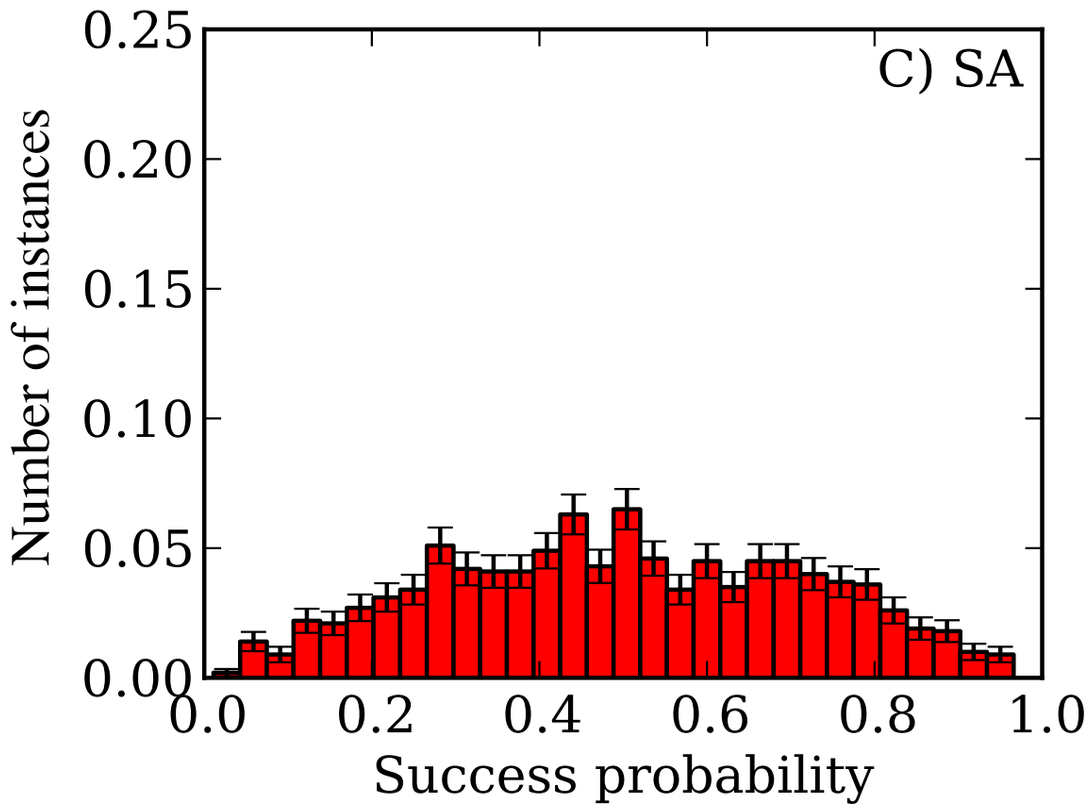
Success probability histograms

1. Pick a specific instance of the couplings J_{ij} and fields h_i

2. Perform N = 1000 or more annealing runs and measure the final energy
   - count the number of times S that we find a ground state
   - calculate the success probability $s = S/N$ of finding a ground state in one run

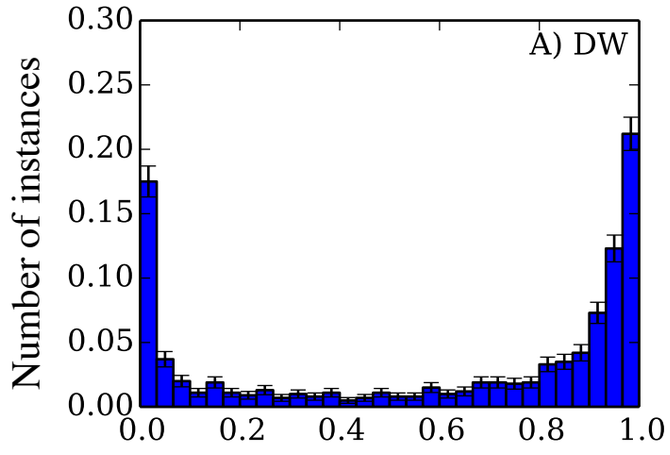
3. Repeat for many instances of the couplings J_{ij} and fields h_i

4. Make a histogram of the success probabilities s



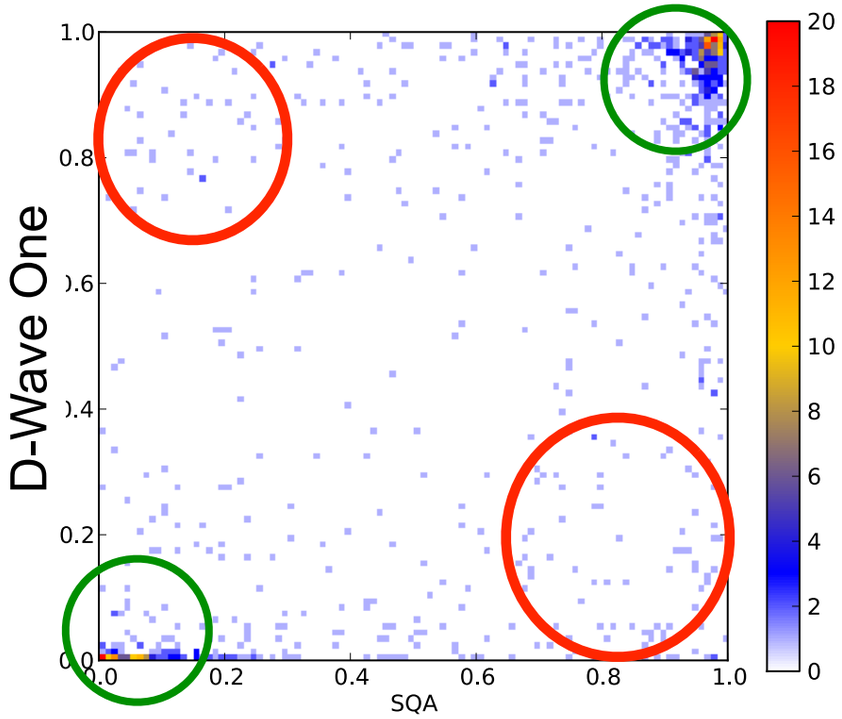
Comparing the histograms

D-Wave One



- Bimodal histogram for D-Wave One and the simulated quantum annealer
  - D-Wave One is **inconsistent with a classical annealer**
  - D-Wave One is **consistent with a simulated quantum annealer**
  - D-Wave One does not look too similar to mean field spin dynamics

Correlations

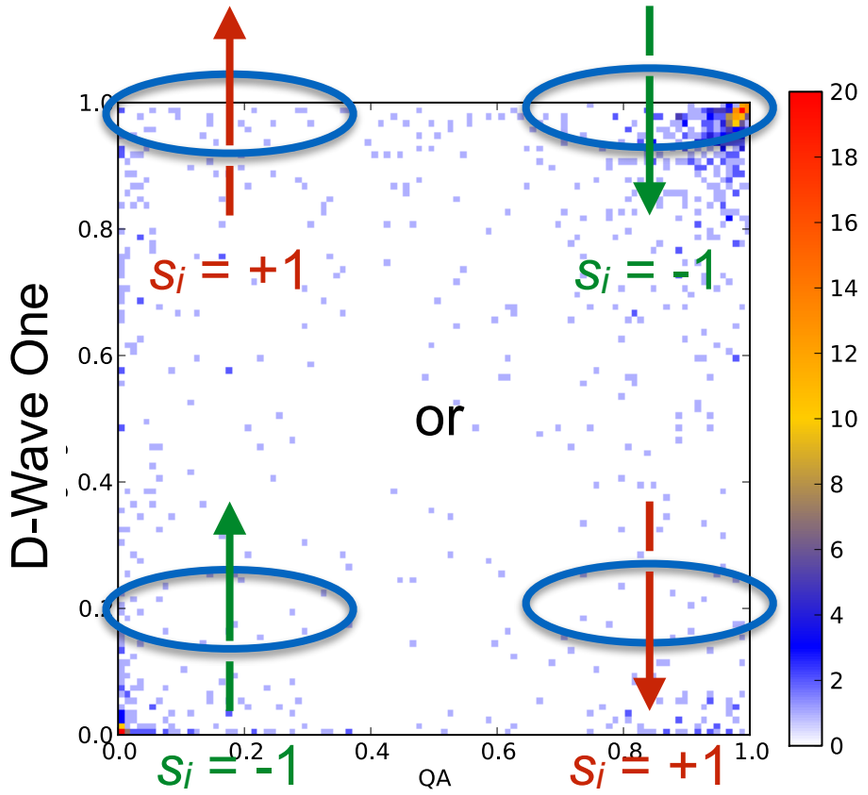


excellent correlations

but 5% outliers:

calibration problems?

Simulated quantum annealer



D-Wave One, gauge transformed

Investigate calibration issues by using a gauge transformation

$$H = \sum_{i,j} J_{ij}\sigma_i\sigma_j$$

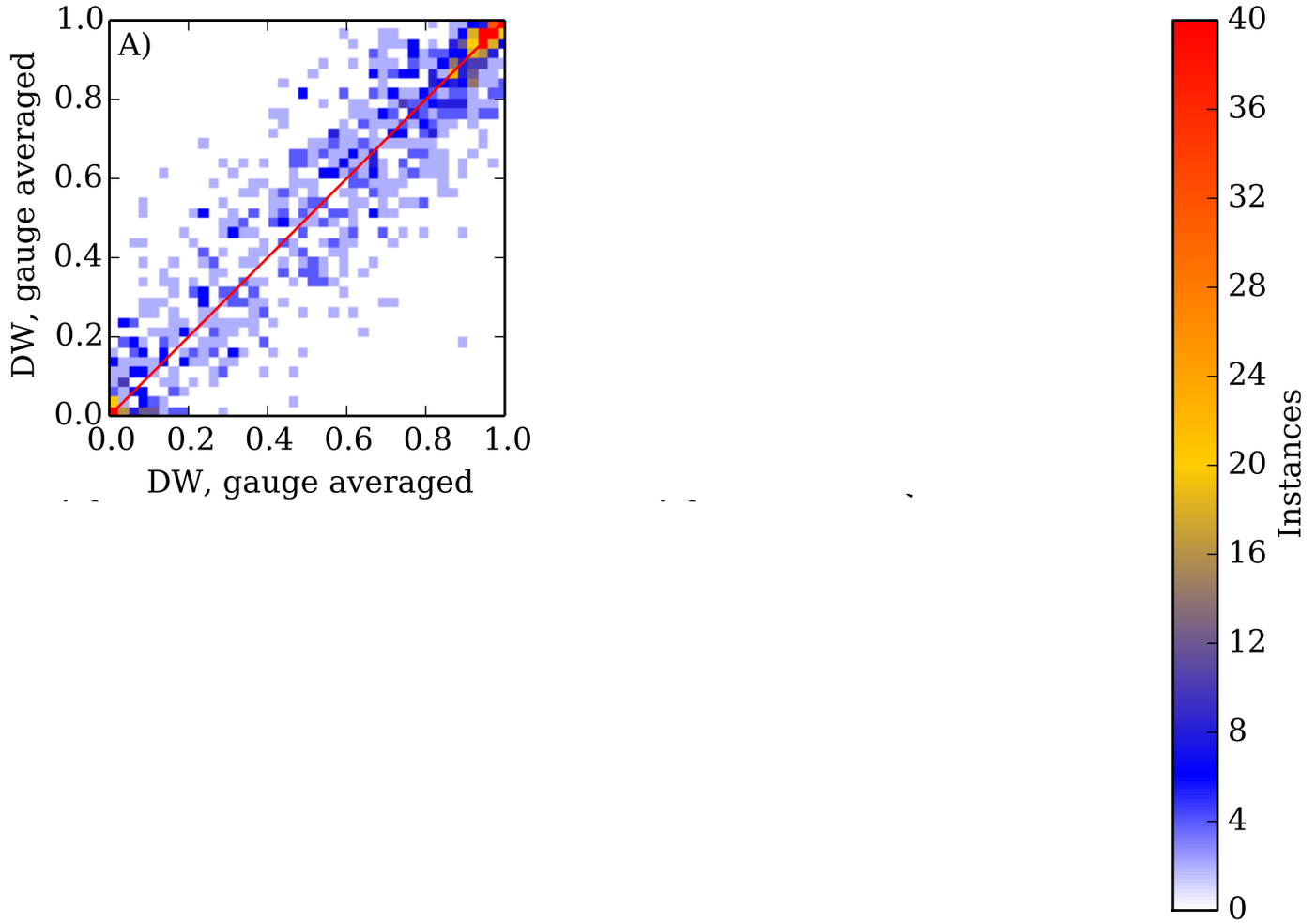
$$\sigma_i \leftarrow a_i\sigma_i \quad \text{with } a_i = \pm 1$$

$$J_{ij} \leftarrow a_i a_j J_{ij}$$

The correlation between a simulated quantum annealer and D-Wave is as good as the correlation of D-Wave with itself

Correlations

D-Wave



simulated quantum annealer

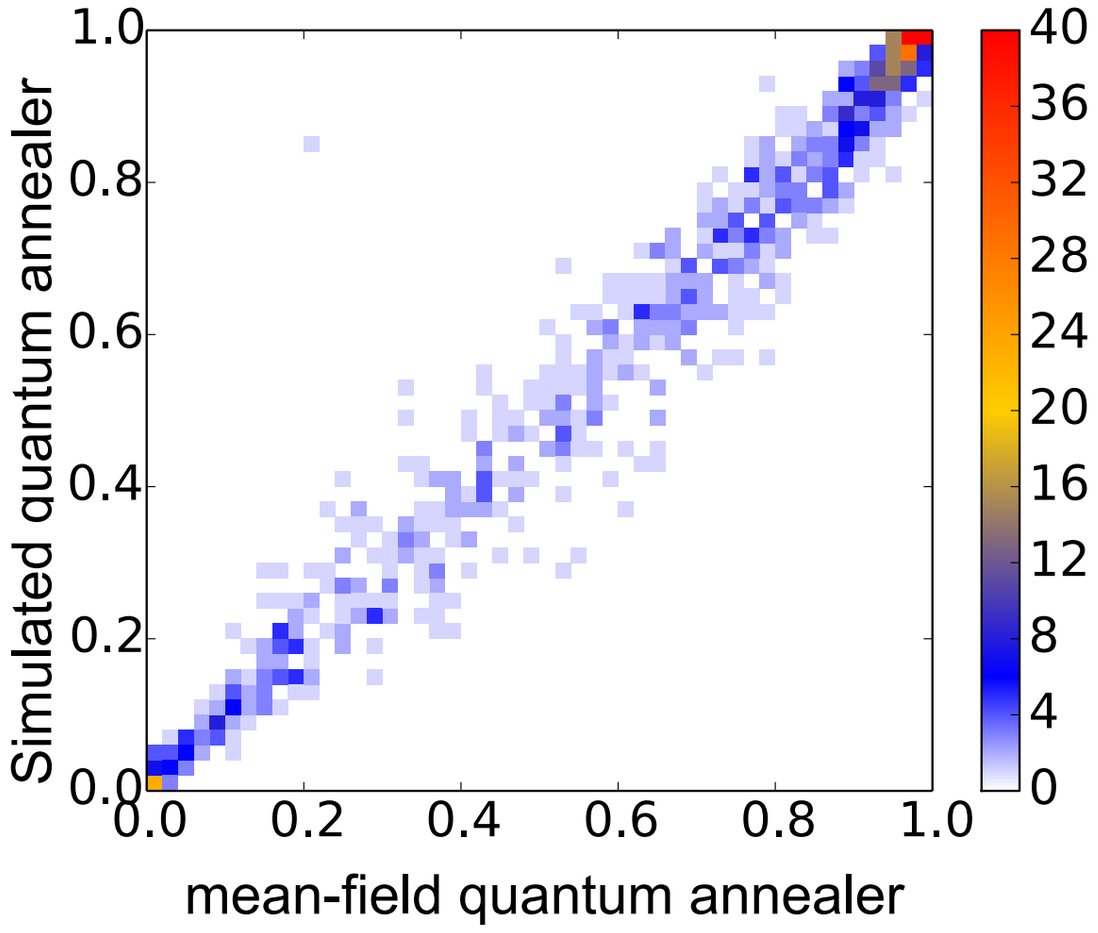
classical annealer

classical spin dynamics

The same instances are hard and easy
on D-Wave and the simulated quantum annealer
but not on D-Wave and mean-field spin dynamics or classical annealing

D-Wave performs like a quantum annealer, but …

- …the simulated quantum annealer has a mean-field version
  Shin, Smolin, Smith and Vazirani, arXiv:1401.7087

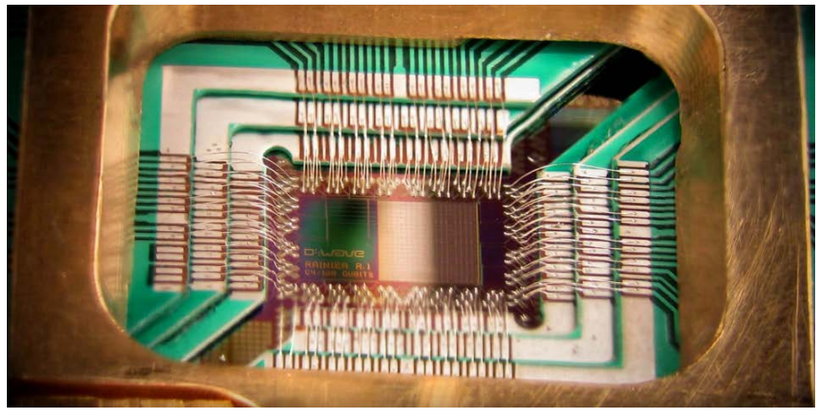


- A quantum annealer at the temperatures where D-Wave operates might not profit much from quantum effects ???

Lessons learned from the D-Wave devices

- There is evidence for entanglement
  Lanting *et al*, PRX (2014)

- There is evidence for collective tunneling
  Boixo *et al*, arXiv:1411.4036

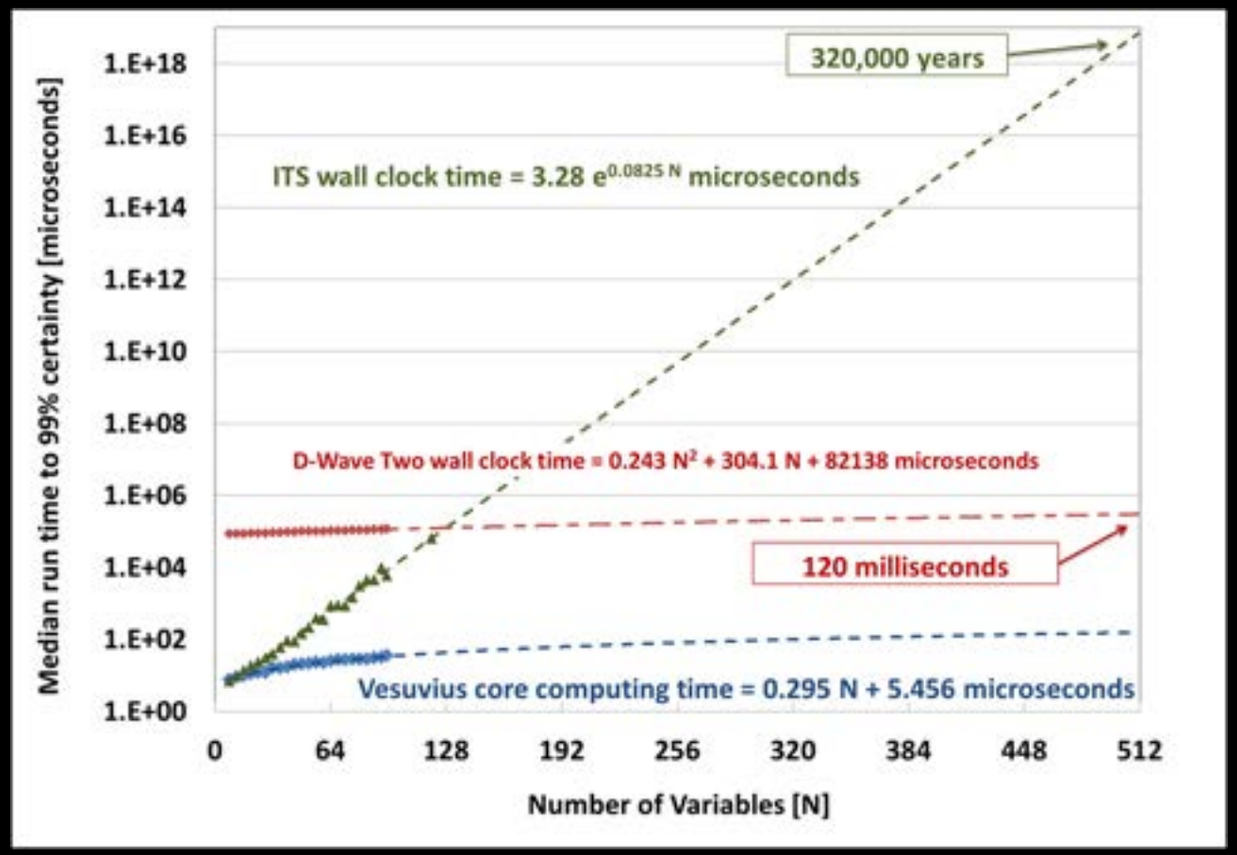


- Performance on spin glasses is consistent with quantum annealing
  Boixo *et al*, Nature Physics (2014)

- But there is also a semi-classical mean-field model that describes the performance well for spin glass instances
  Shin *et al*, arXiv:1401.7087

- What about quantum speedup?

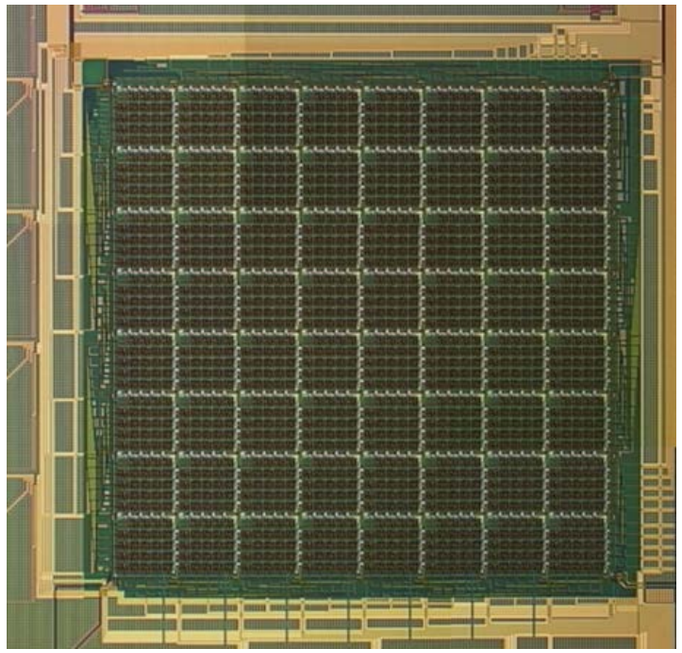
The ultimate question: how does it scale?



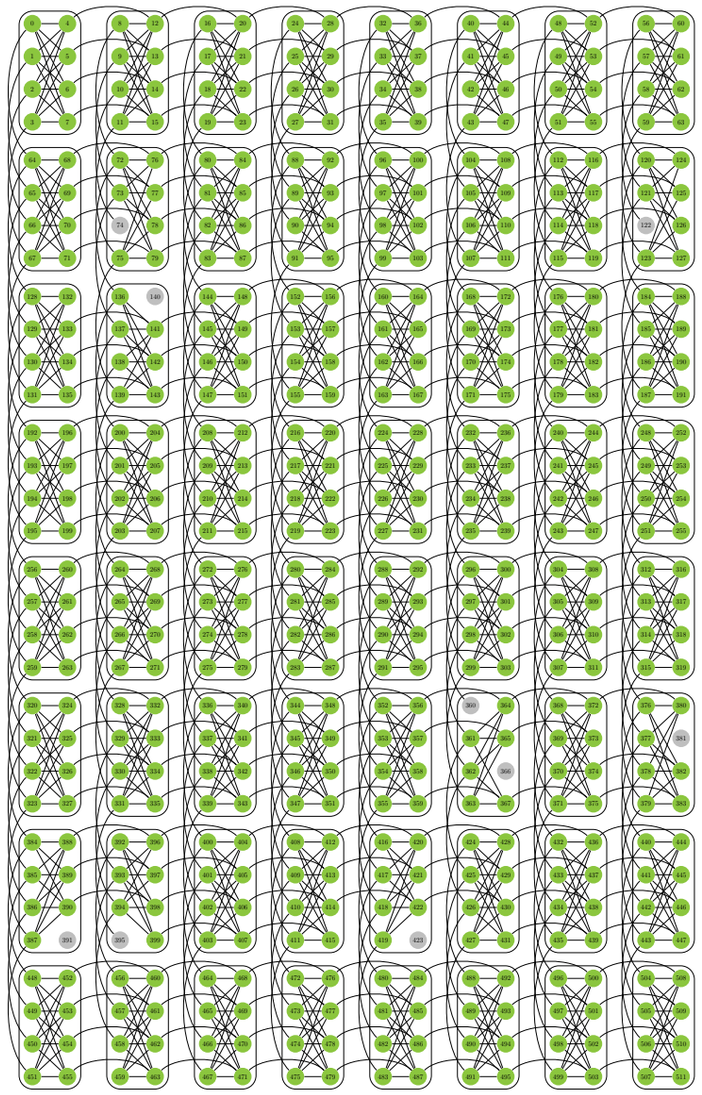
Our experiments

Find test problems for the machine to solve

random ±1 couplings on all bonds of the chimera graph



Quantum annealing on D-Wave Two

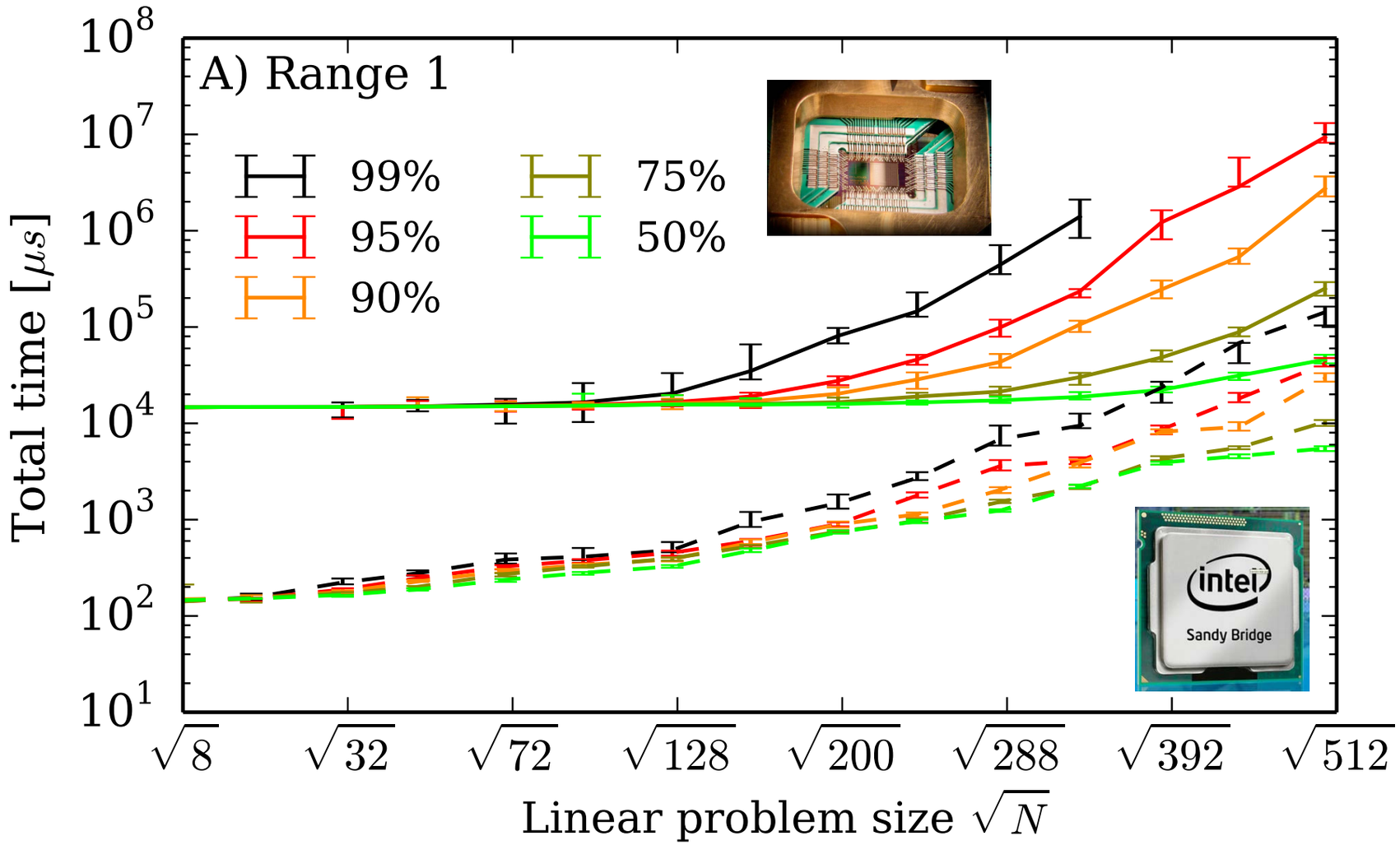


Simulated annealing on Intel CPU

Codes are on arXiv:1401.1084

Scaling of wall-clock times

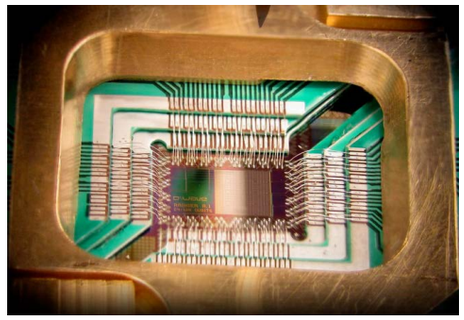
- Time to find the ground state with 99% probability
  D-Wave compared to a simulated annealer on a CPU

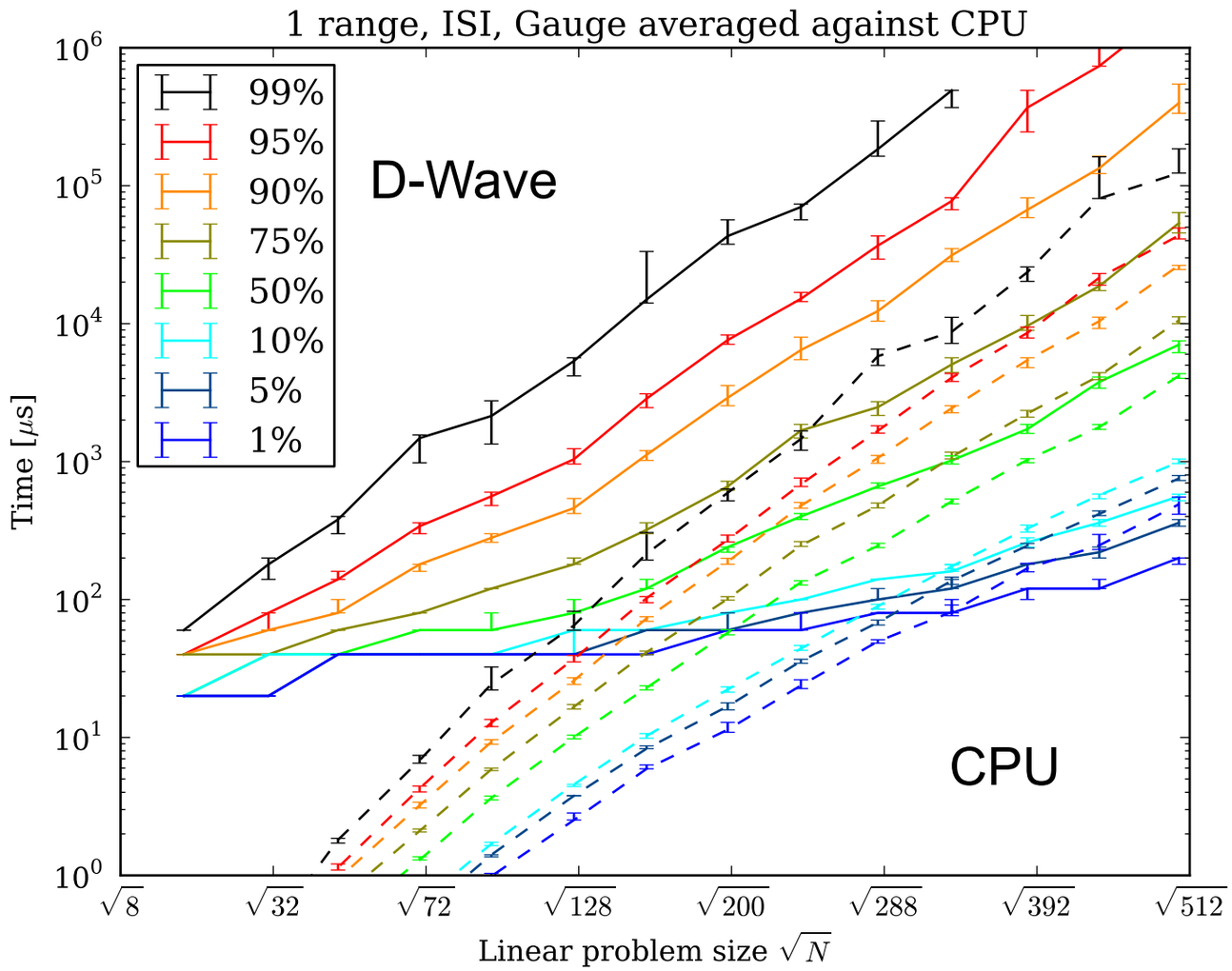


- Programming overhead dominates for small problems

Scaling of pure annealing times

To extrapolate to larger sizes focus just on annealing time to get the intrinsic scaling





Detecting and defining quantum speedup

…. a non-trivial endeavor

Defining quantum speedup

Quantum speedup exists if

$$S(N) = \frac{T_C(N)}{T_Q(N)}$$

grows with the problem size N

Seems easy and trivial to define, but …

one can easily get fooled into believing there is speedup

Five types of quantum speedup

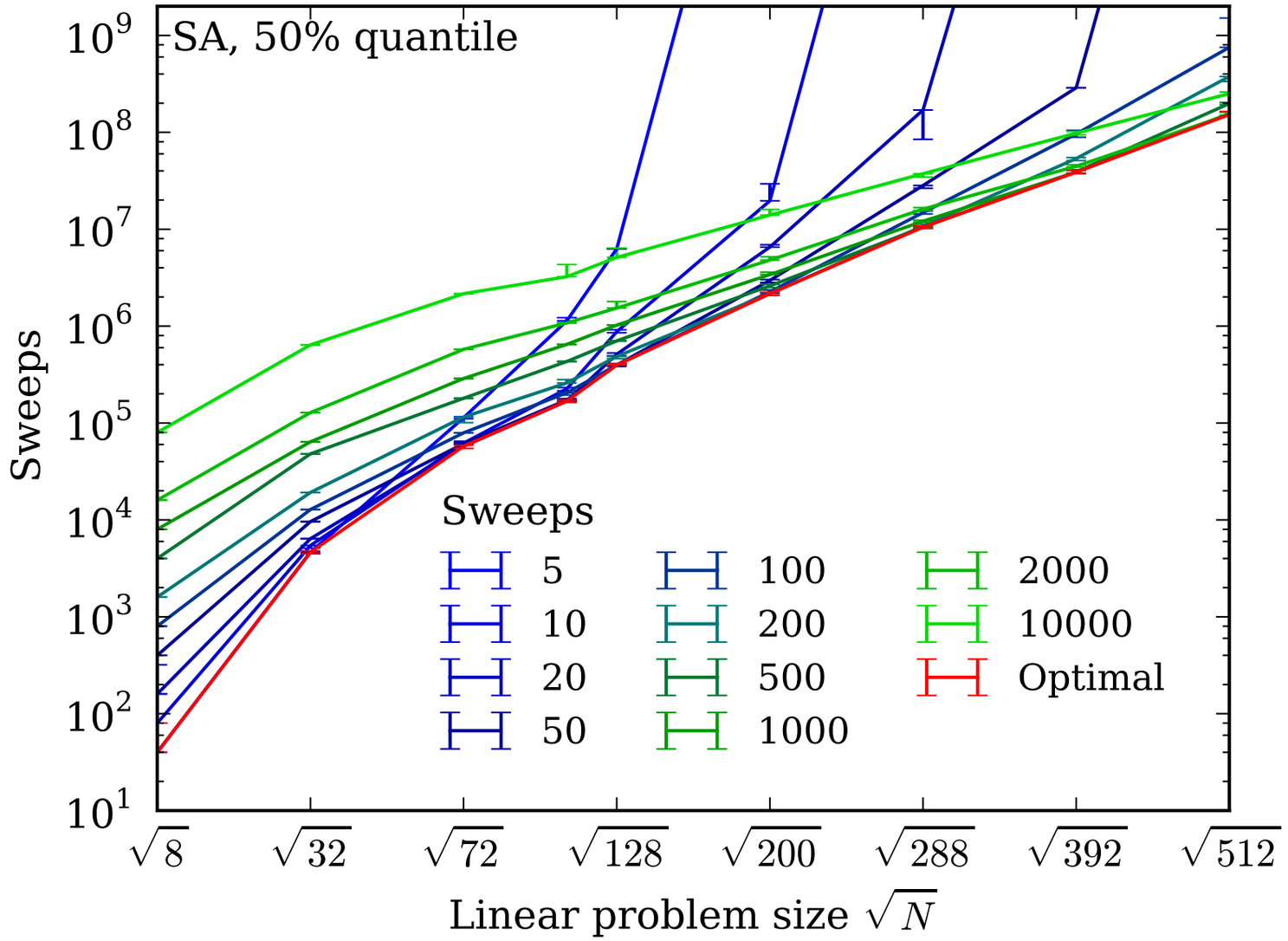
- **Provable quantum speedup**
  - when we can prove a separation between T_Q and T_C
  - example: Grover search
- **Strong quantum speedup** (Traub *et al*, 2013)
  - speedup compared to bound for **best classical algorithm**, whether that algorithm is known or not
- **Quantum speedup**
  - speedup compared to **best known** classical algorithm
  - example: Shor's algorithm
- **Potential (quantum) speedup**
  - speedup compared to a (selection of) classical algorithms
- **Limited quantum speedup**
  - speedup compared to a "classical version" of the quantum algorithm

One can easily get fooled …

- Suboptimal performance at smaller sizes look like speedup
- Parallel speedup can be mistaken for quantum speedup
- …

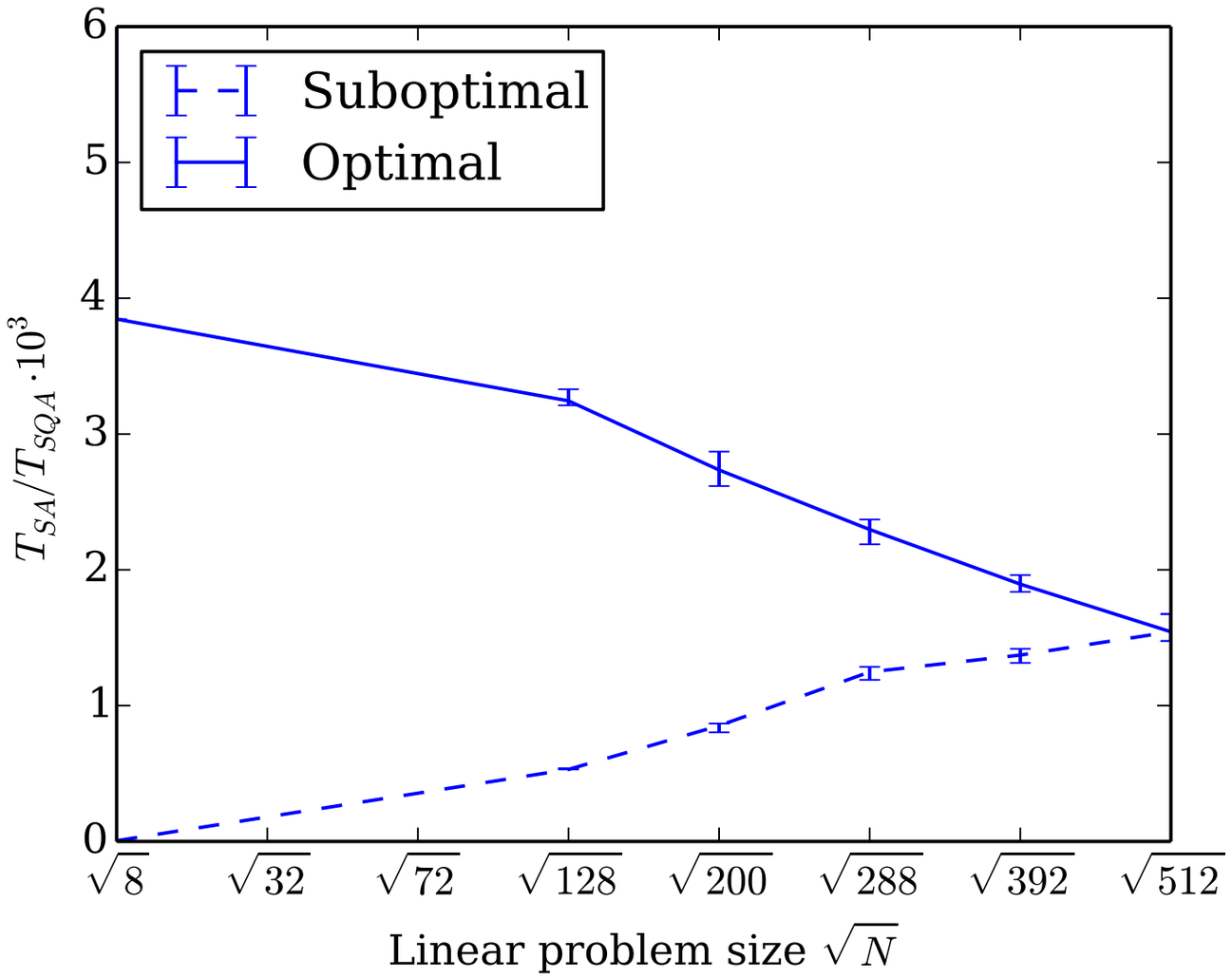
Performance at fixed annealing time

- Initially too flat slope when running at a fixed annealing time
- To determine asymptotic scaling we have to find the optimal annealing time for each problem size



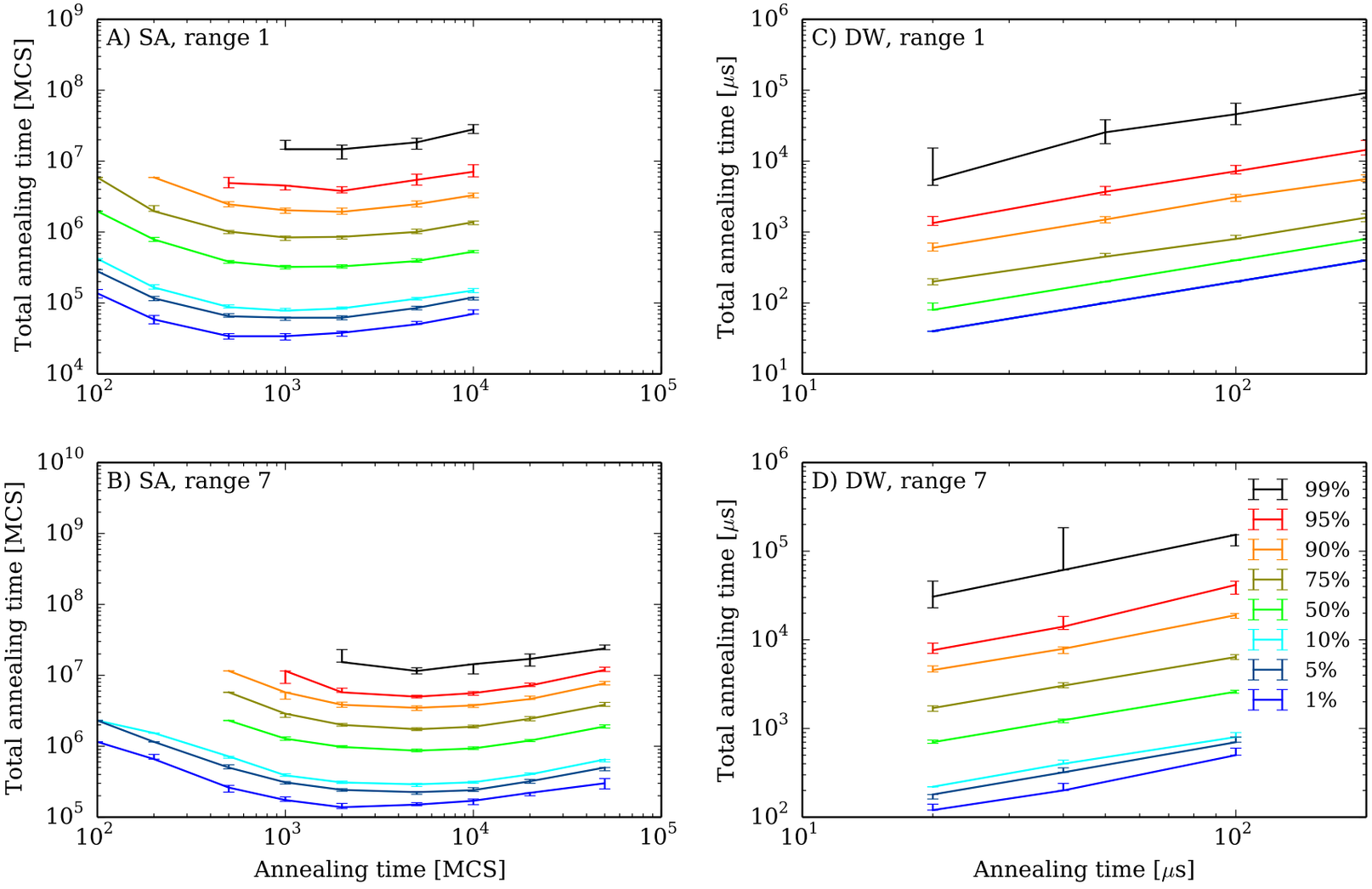
“Fake” speedup due to suboptimal performance

- Compare simulated quantum annealing at fixed (suboptimal) annealing time to classical annealing at optimal annealing time.
- What is a slowdown suddenly looks like speedup



Optimizing the total effort

- On DW2 the optimal annealing time is much shorter than 20 µs.
  - The annealing times are far longer than is needed
  - The machine could be much faster
  - We cannot demonstrate quantum speedup without doubt

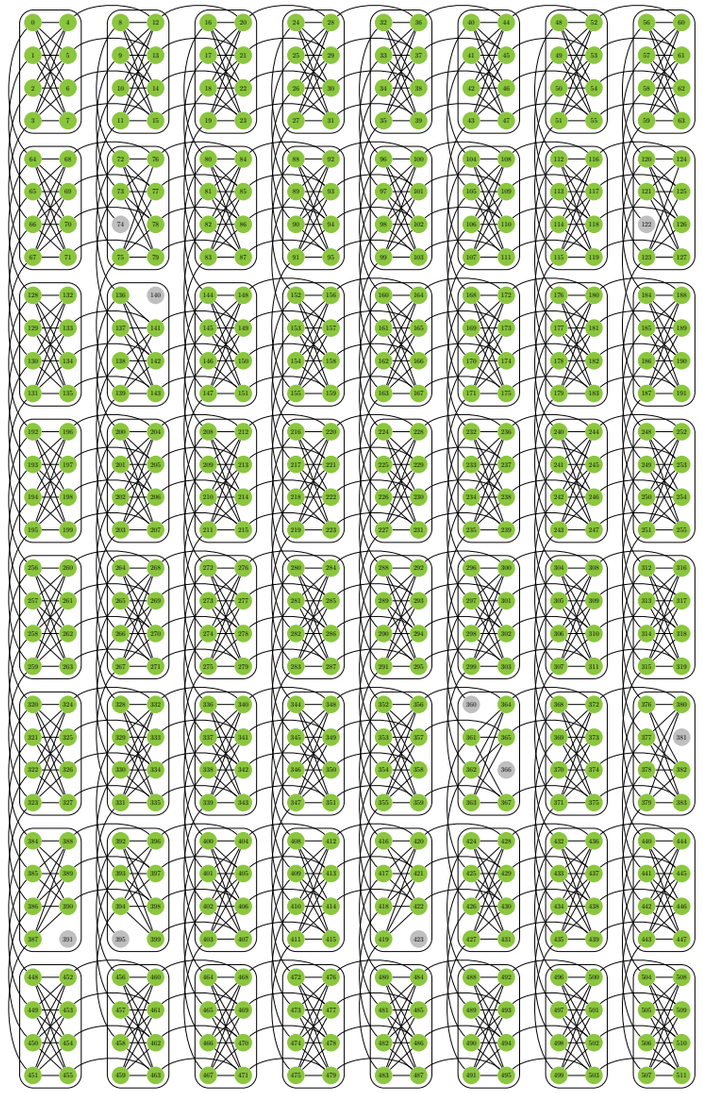


Parallel versus quantum speedup

- D-Wave Two is a parallel machine acting on all spins simultaneously
- We need to compare to a parallel classical machine with the same hardware layout

$$T_{\mathrm{C}}(N) \propto \frac{1}{N} T_{\mathrm{SA}}(N)$$

$$S(N) = \frac{T_{\mathrm{C}}(N)}{T_{\mathrm{DW}}(N)} \propto \frac{T_{\mathrm{SA}}(N)}{T_{\mathrm{DW}}} \frac{1}{N}$$

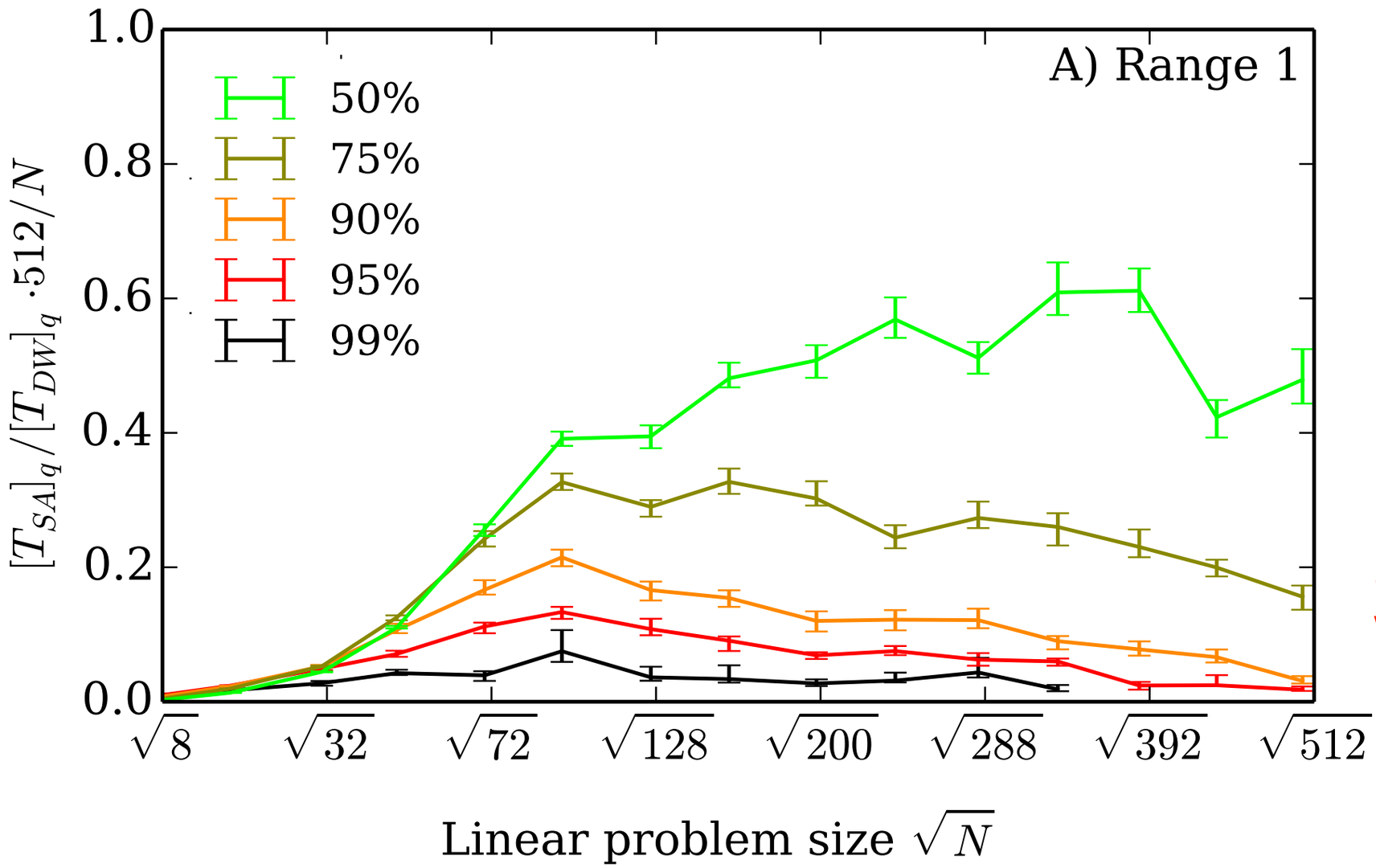


- One can easily build such a classical machine in an FPGA or special purpose chip (ASIC)

[1]

Is there quantum speedup?

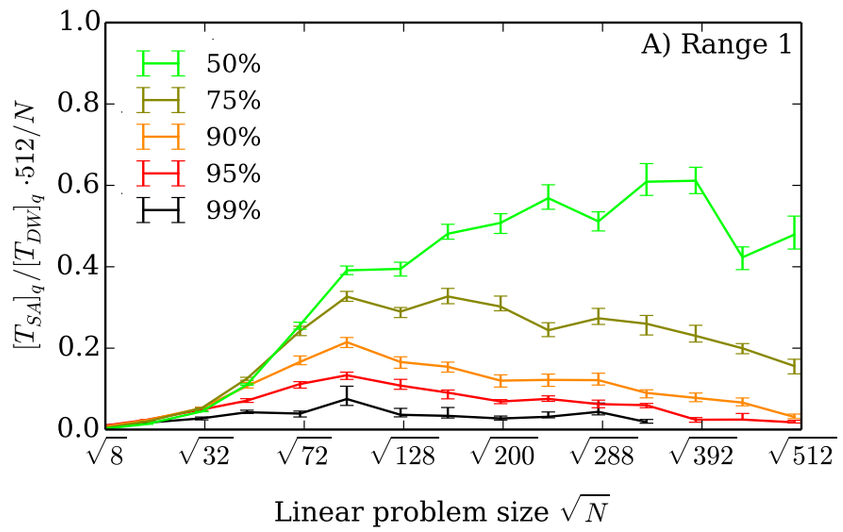
Compare scaling of D-Wave with simulated annealing



Same performance if we want to solve half of the problems

Slowdown if we want to solve more

Why is there no quantum speedup?

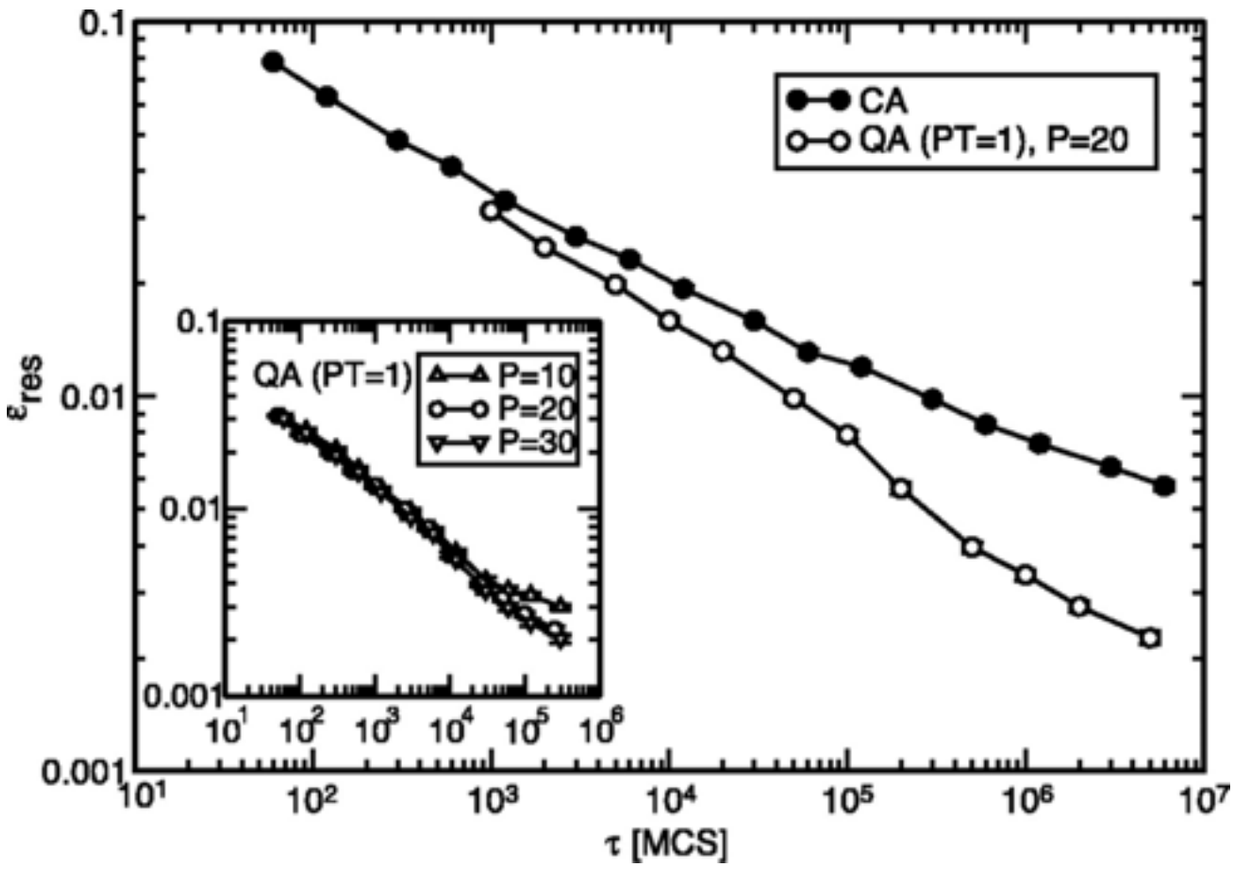


- Is it due to the 5% calibration errors and we simply solve the wrong problem instance?
- Are spin glasses the wrong problem for quantum annealers and we need other problems to see quantum speedup?
- Is it due to limited coherence and non-zero temperature?
- Why did previous QMC simulations [Santoro *et al,* Science (2002)] give evidence for quantum annealing outperforming classical annealing?

Towards a better understanding

Simulated classical versus quantum annealing

- Early evidence for superiority of QA came from QMC simulations
- Lower residual energies in simulated quantum annealing compared to simulated thermal annealing, but not for all models



Santoro *et al,* Science (2002)

similar results by

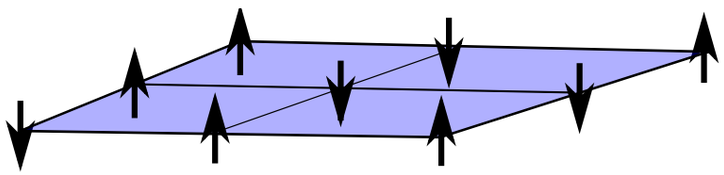
Matsuda, Nishimori, Katzgraber (2009)

this is inconsistent with what we saw on D-Wave!

From classical to quantum Monte Carlo

- Simulated annealing performs a Monte Carlo simulation
- Sample random configurations according to their Boltzmann weights

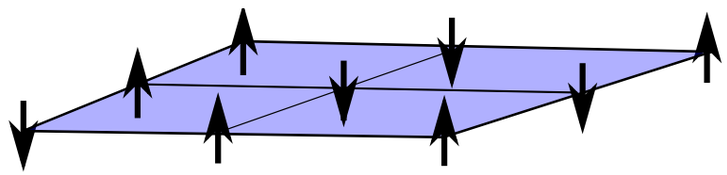
$$Z = \sum_{s_1,\ldots,s_N} \exp(-\beta \sum_{i<j} J_{ij} s_i s_j)$$



From classical to quantum Monte Carlo

- Simulated quantum annealing performs a quantum Monte Carlo simulation
- Sample random configurations according to their Boltzmann weights???

$$Z = \mathrm{Tr}\exp(-\beta H) \neq \sum_c \exp(-\beta E_c)$$



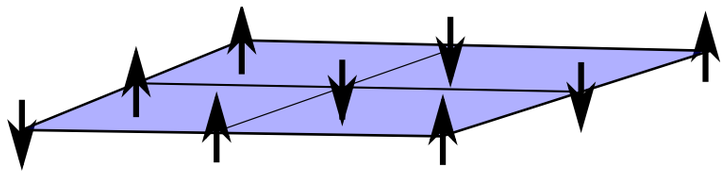
From classical to quantum Monte Carlo

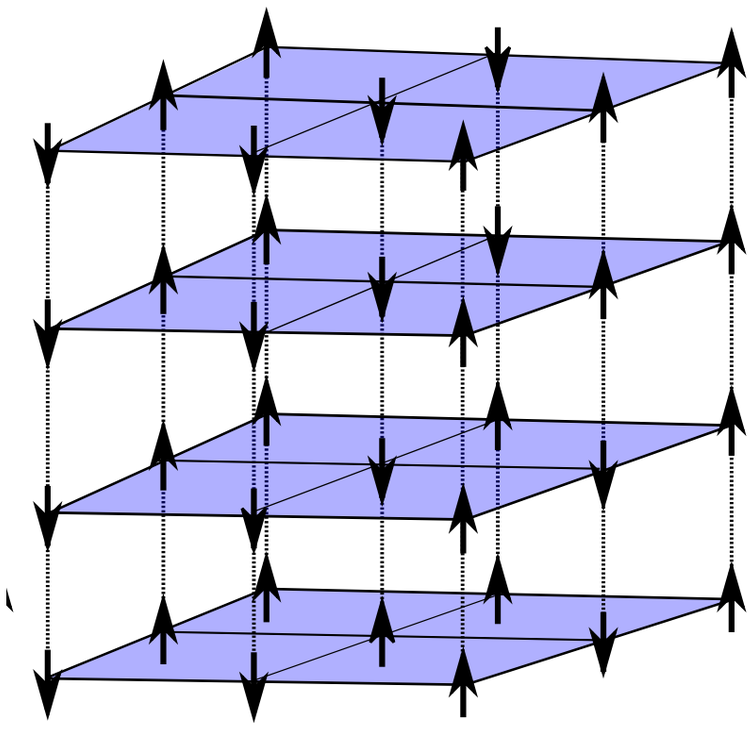
- Simulated quantum annealing performs a quantum Monte Carlo simulation
- Map quantum system to classical path integral
- Sample random configurations according to their weights

$$Z = \mathrm{Tr}\exp(-\beta H) \neq \sum_c \exp(-\beta E_c)$$

$$Z = \mathrm{Tr}\exp(-\beta H) = \mathrm{Tr}\exp[(-\Delta_\tau H)^M]$$

$$\approx \sum_{(s_{i,\tau})} \exp\left(-\Delta_\tau \sum_{i<j} \sum_{\tau} J_{ij} s_{i,\tau} s_{j,\tau} - \Delta_\tau \sum_{i} \sum_{\tau} J' s_{i,\tau} s_{i,\tau+1} \right) + O(\Delta_\tau^2)$$



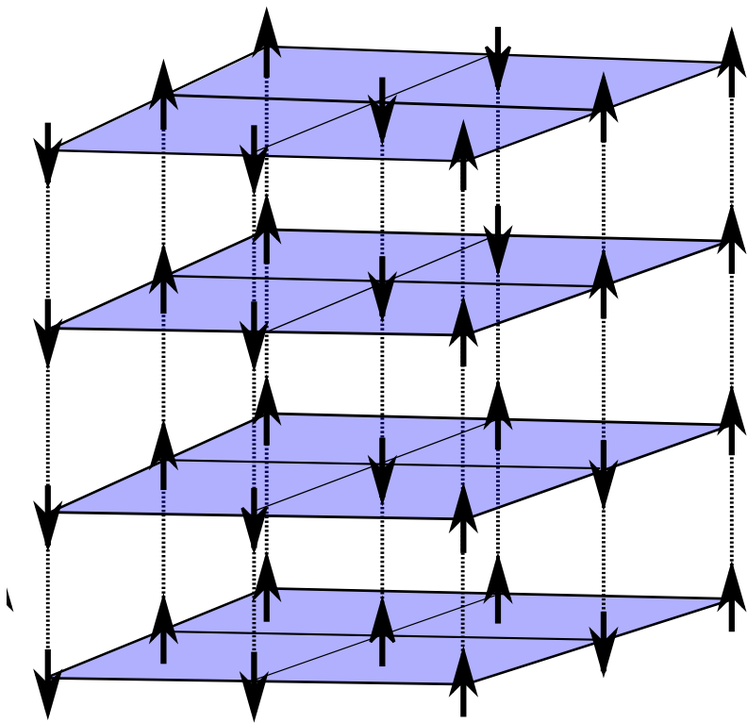


From classical to quantum Monte Carlo

- Simulated quantum annealing performs a quantum Monte Carlo simulation
- Map quantum system to classical path integral
- Sample random configurations according to their weights

$$Z = \mathrm{Tr}\exp(-\beta H) = \mathrm{Tr}\exp[(-\Delta_\tau H)^M]$$

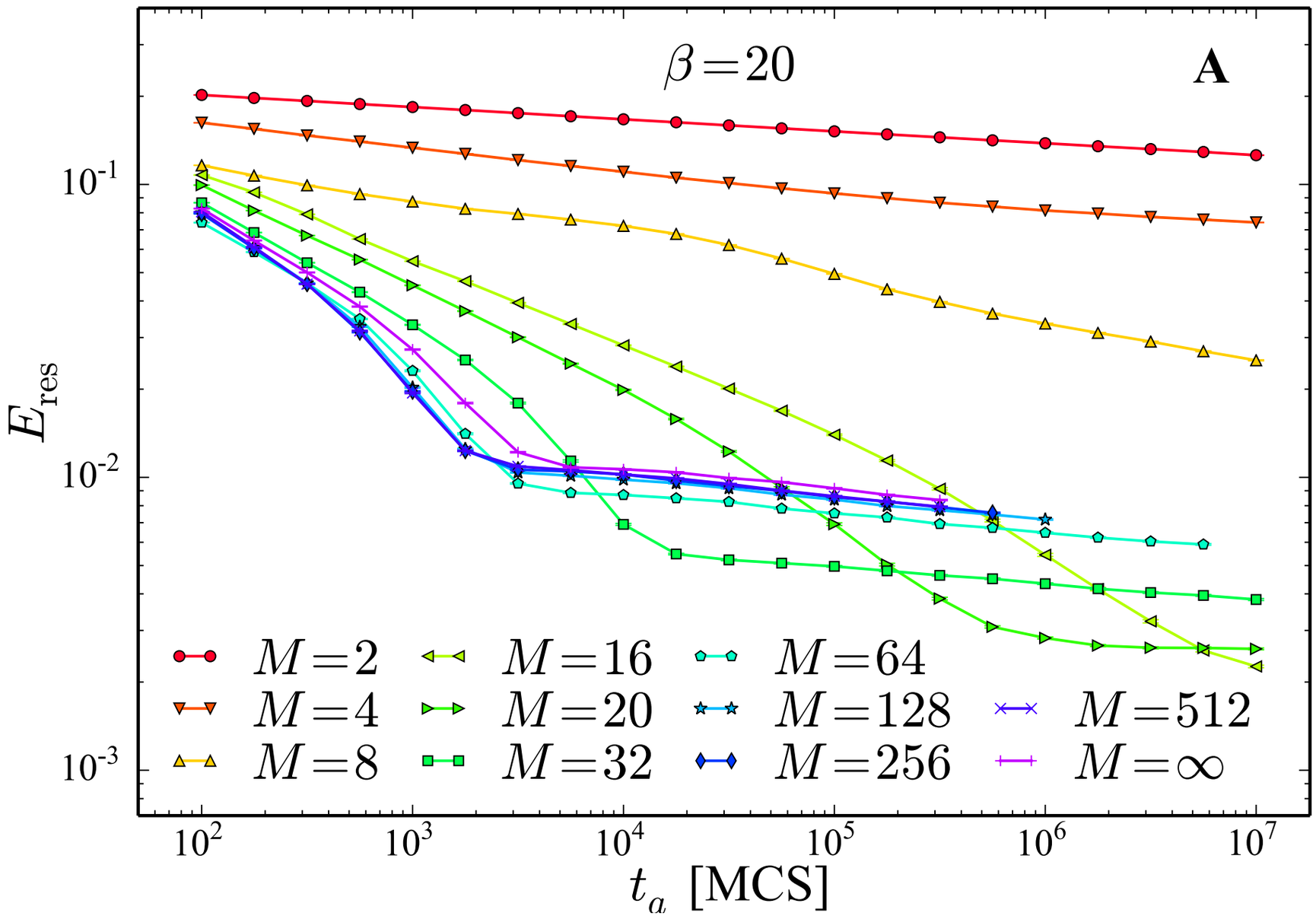
$$\approx \sum_{(s_{i,\tau})} \exp\left(-\Delta_\tau \sum_{i<j} \sum_{\tau} J_{ij} s_{i,\tau} s_{j,\tau} - \Delta_\tau \sum_{i} \sum_{\tau} J' s_{i,\tau} s_{i,\tau+1} \right) + O(\Delta_\tau^2)$$



Time step dependence

Heim *et al*, Science (2015)

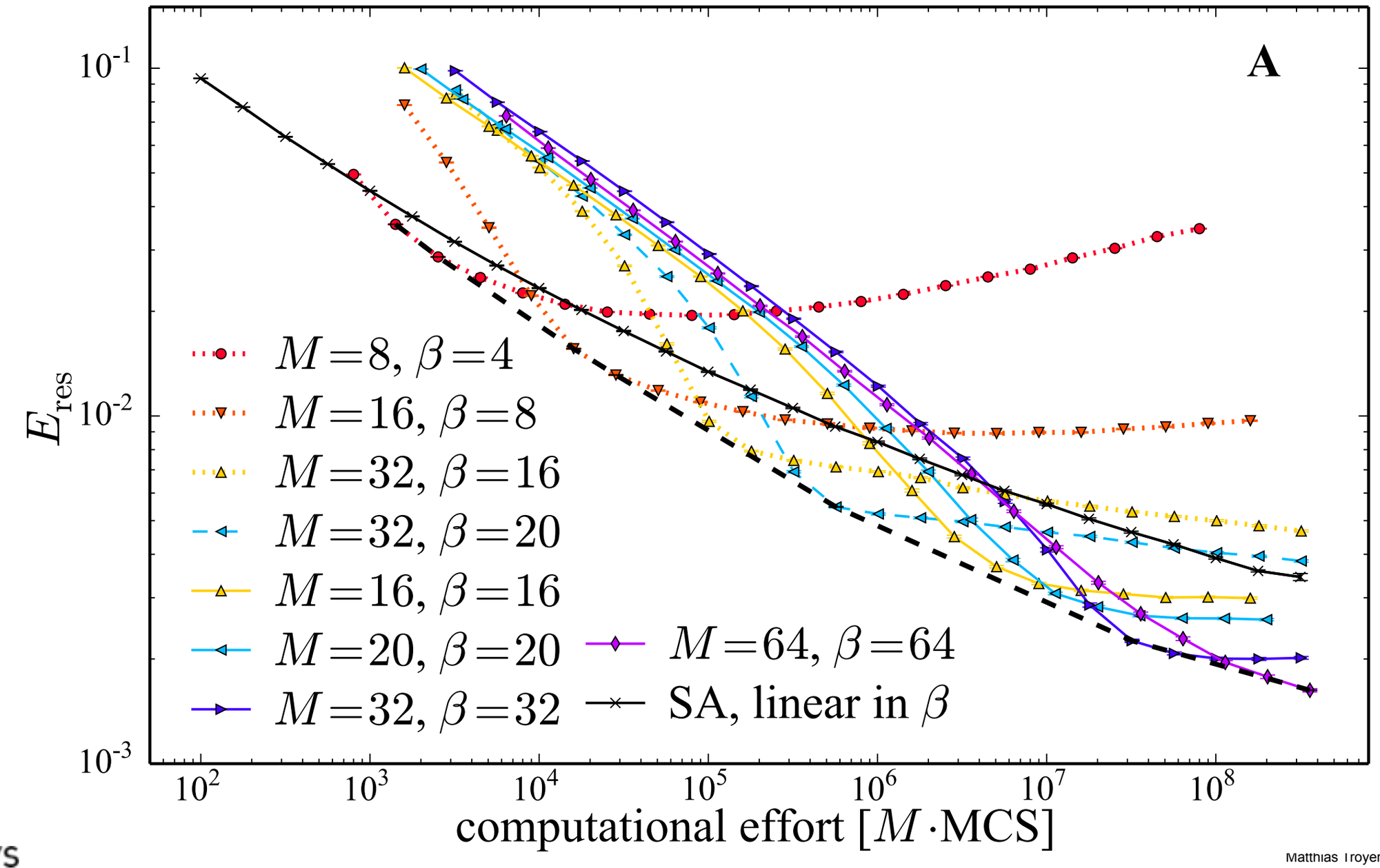
- The behavior of a simulated quantum annealer depends strongly on discretization of imaginary time path integrals



Performance as a classical optimizer

Heim *et al*, Science (2015)

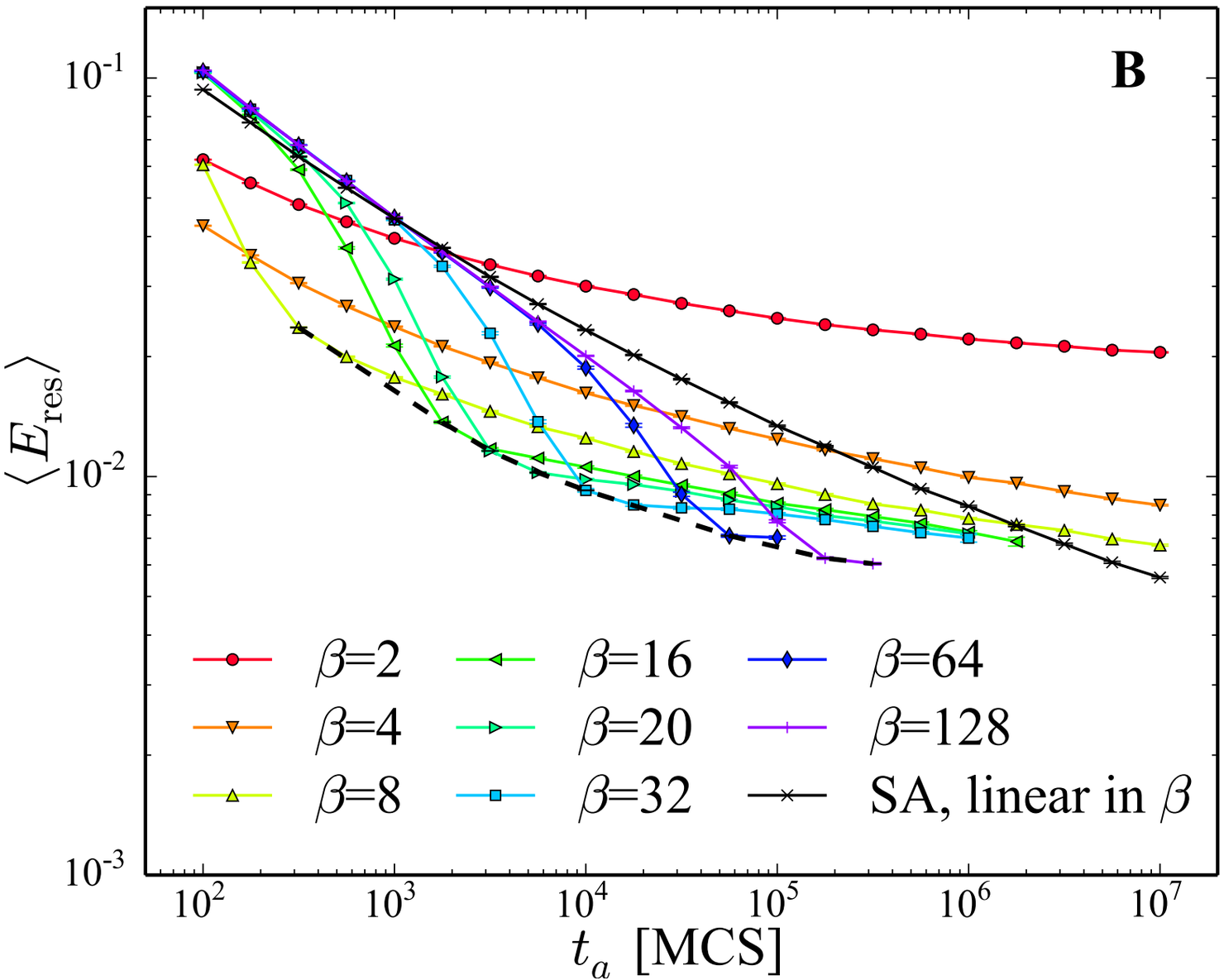
- Use a large time step to get best performance



Performance in the “physical” limit

Heim *et al*, Science (2015)

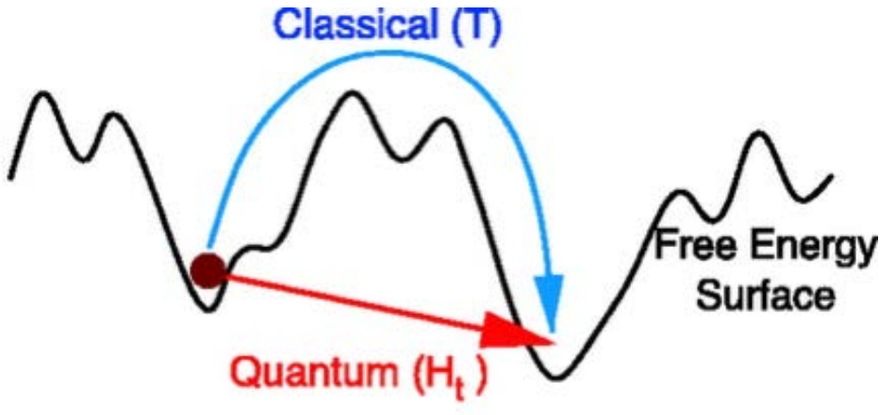
- Faster than classical annealer for short times
- The advantage vanishes for long times
- One should use infinitesimal time steps to make predictions for devices

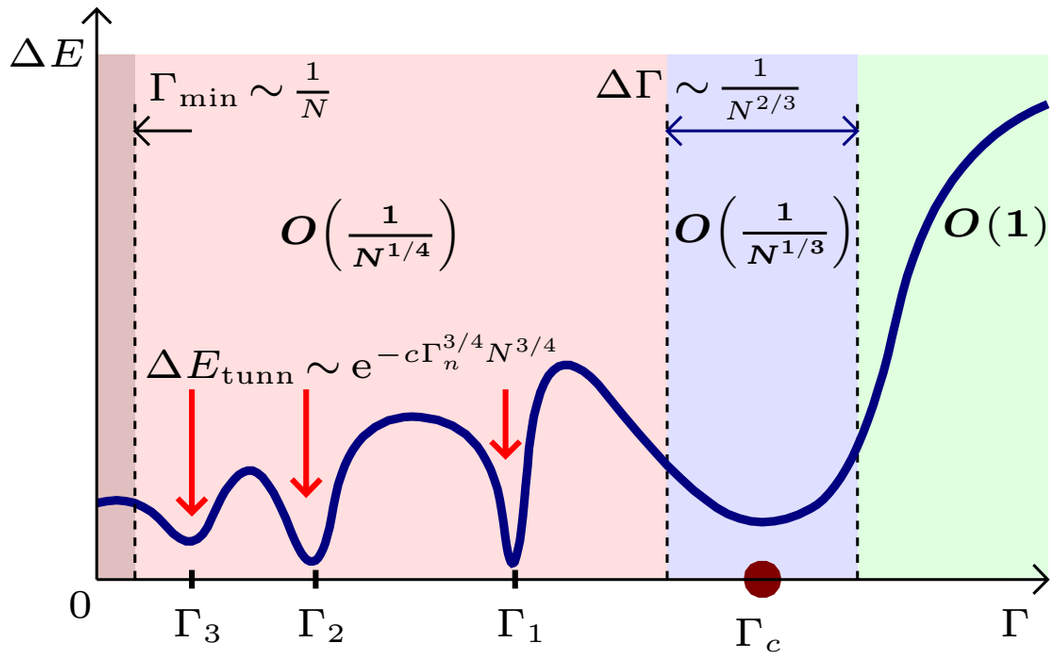


But does QMC tell us anything about the real performance of a quantum annealer?

Scaling of annealing times in quantum annealing

- Quantum adiabatic theorem $t \propto \frac{1}{\Delta_{\min}^2}$
- Open systems tunneling through a barrier $t \propto \epsilon^2 \propto \frac{1}{\Delta_{\min}^2}$





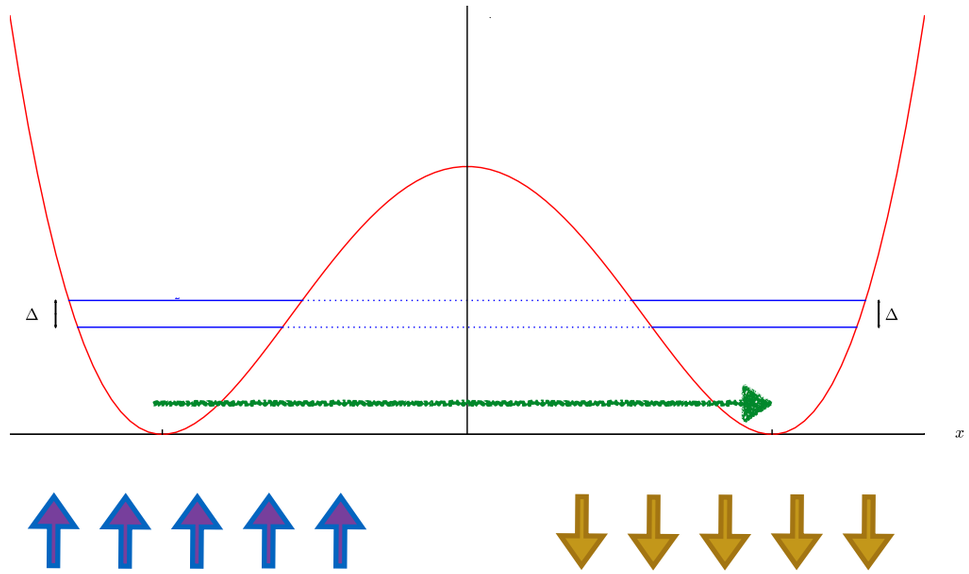
S. Knysh, arXiv:1506.08608

similar results Farhi *et al* (2012)

- How does QMC scale compared to that?

Tunneling between two states of a ferromagnet

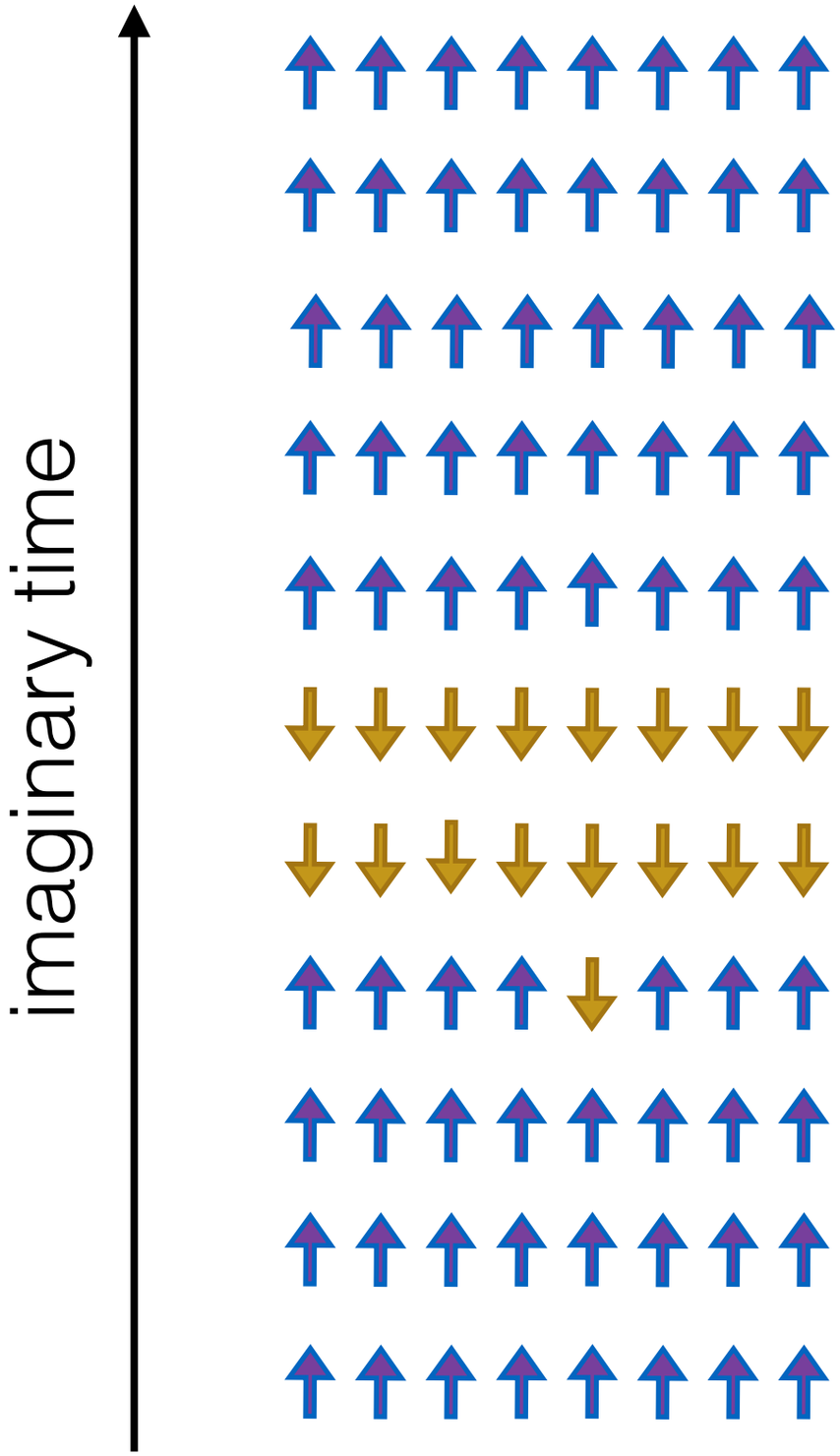
$$H = -\sum_{\langle ij \rangle} J\, \sigma_i^z \sigma_j^z - \Gamma \sum_i \sigma_i^x$$

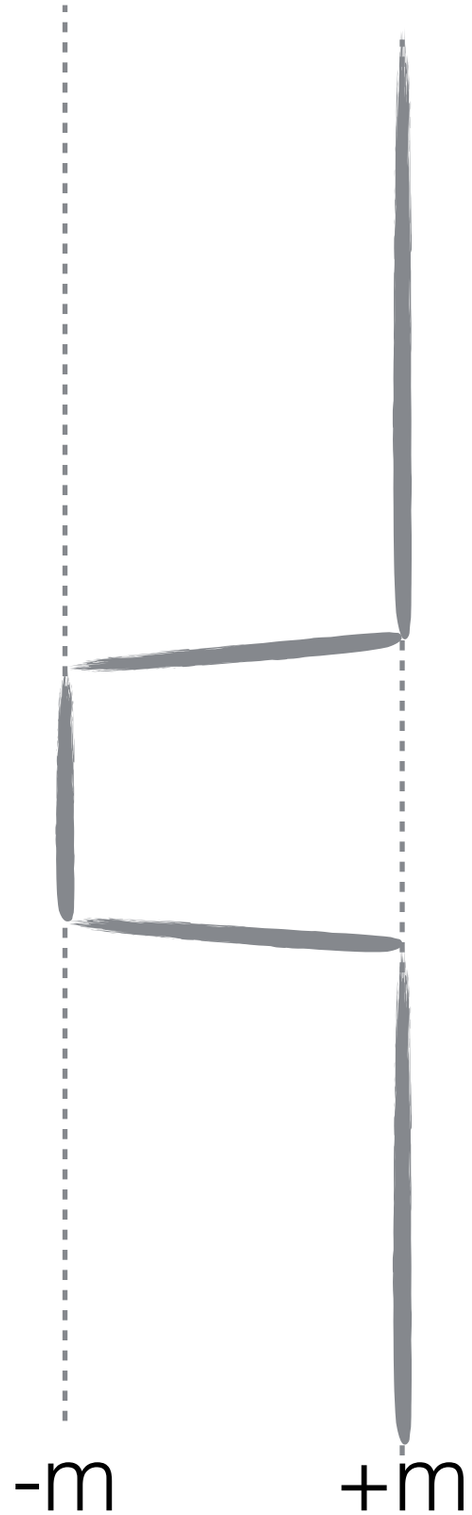


energy splitting Δ

$$\psi_0 = \left| \uparrow\uparrow\uparrow\uparrow\uparrow \right\rangle + \left| \downarrow\downarrow\downarrow\downarrow\downarrow \right\rangle$$

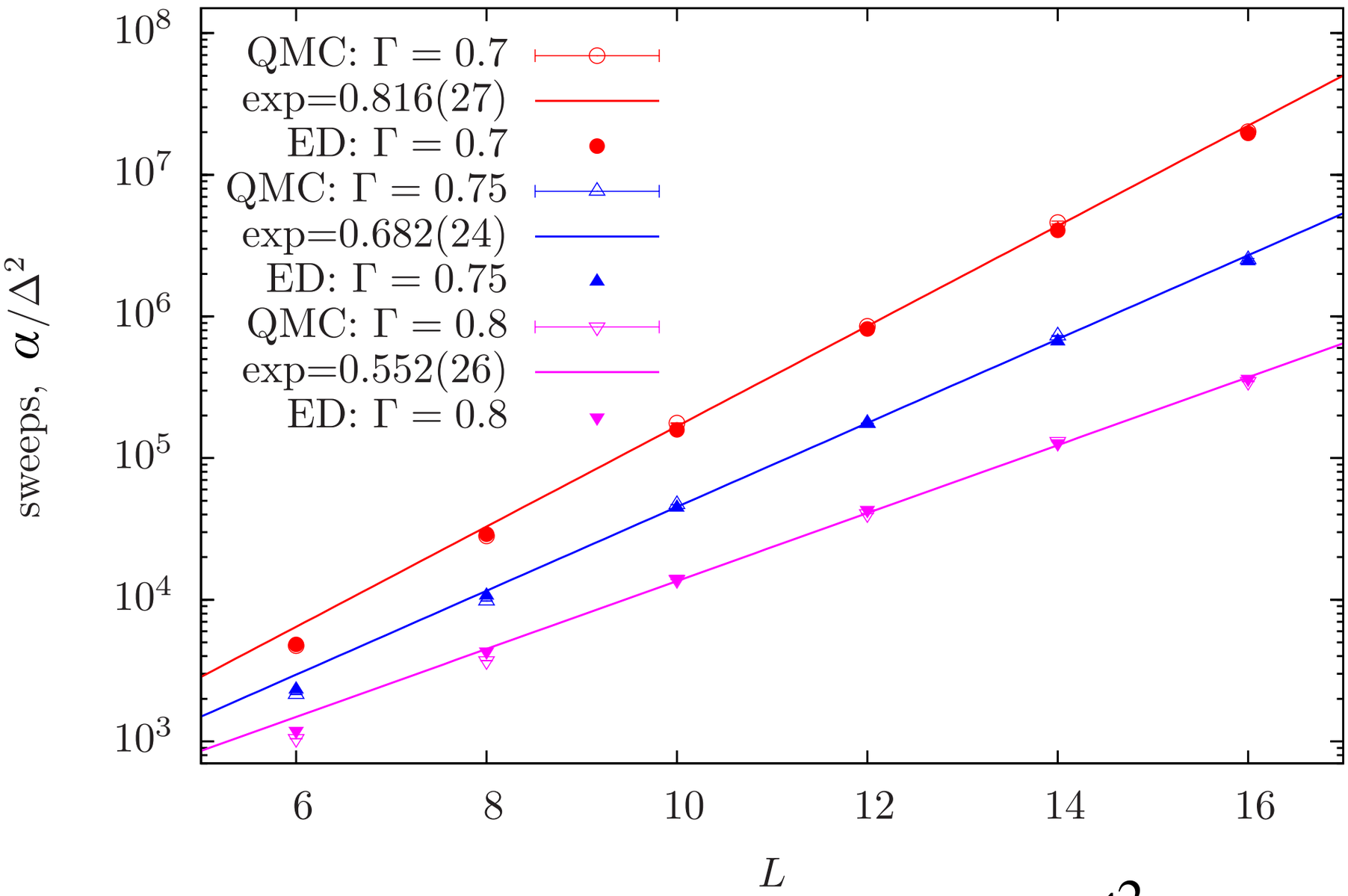
$$\psi_1 = \left| \uparrow\uparrow\uparrow\uparrow\uparrow \right\rangle - \left| \downarrow\downarrow\downarrow\downarrow\downarrow \right\rangle$$





Tunneling determined from the time QMC needs to create an instanton.

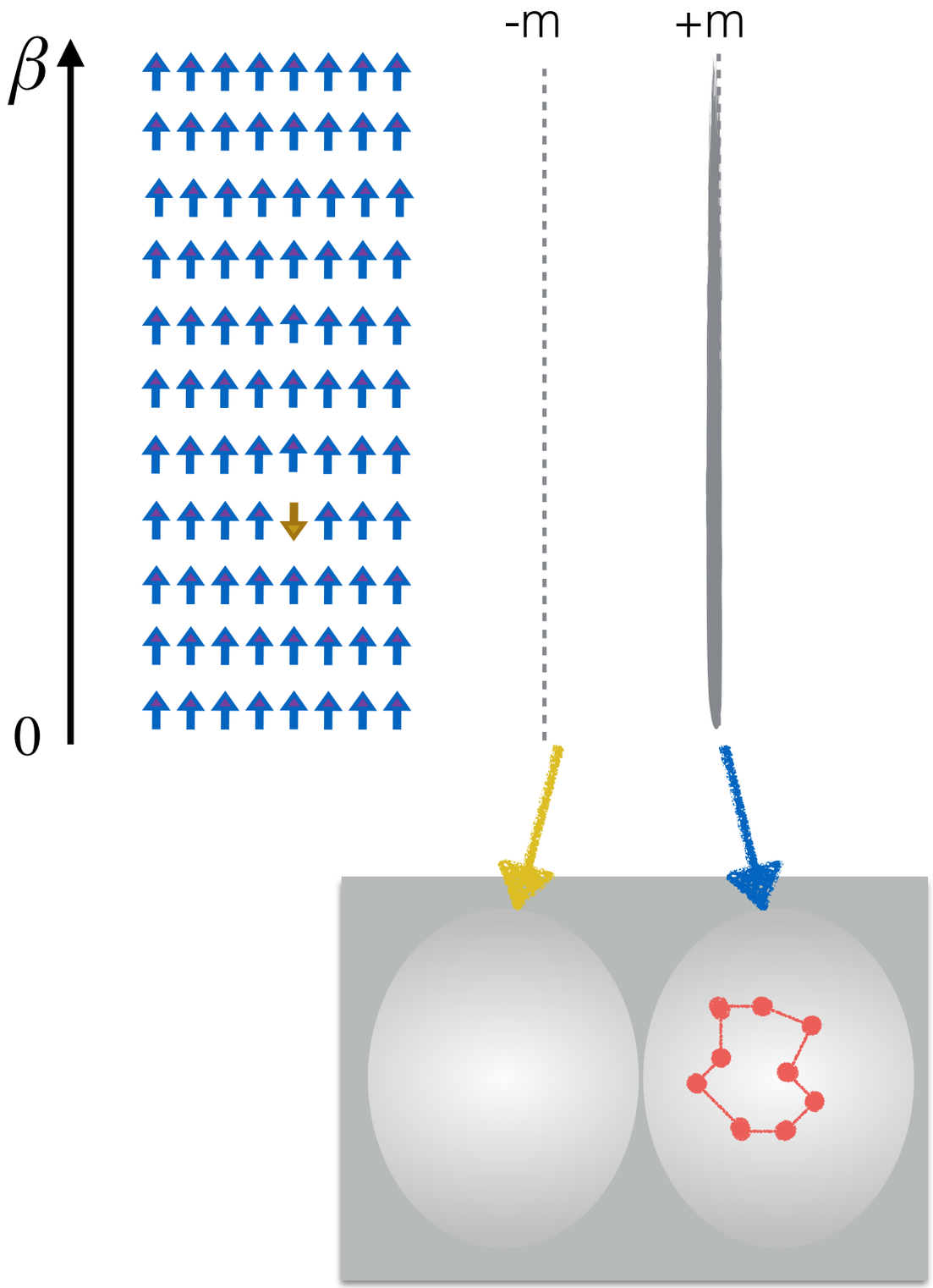
Scaling with system size

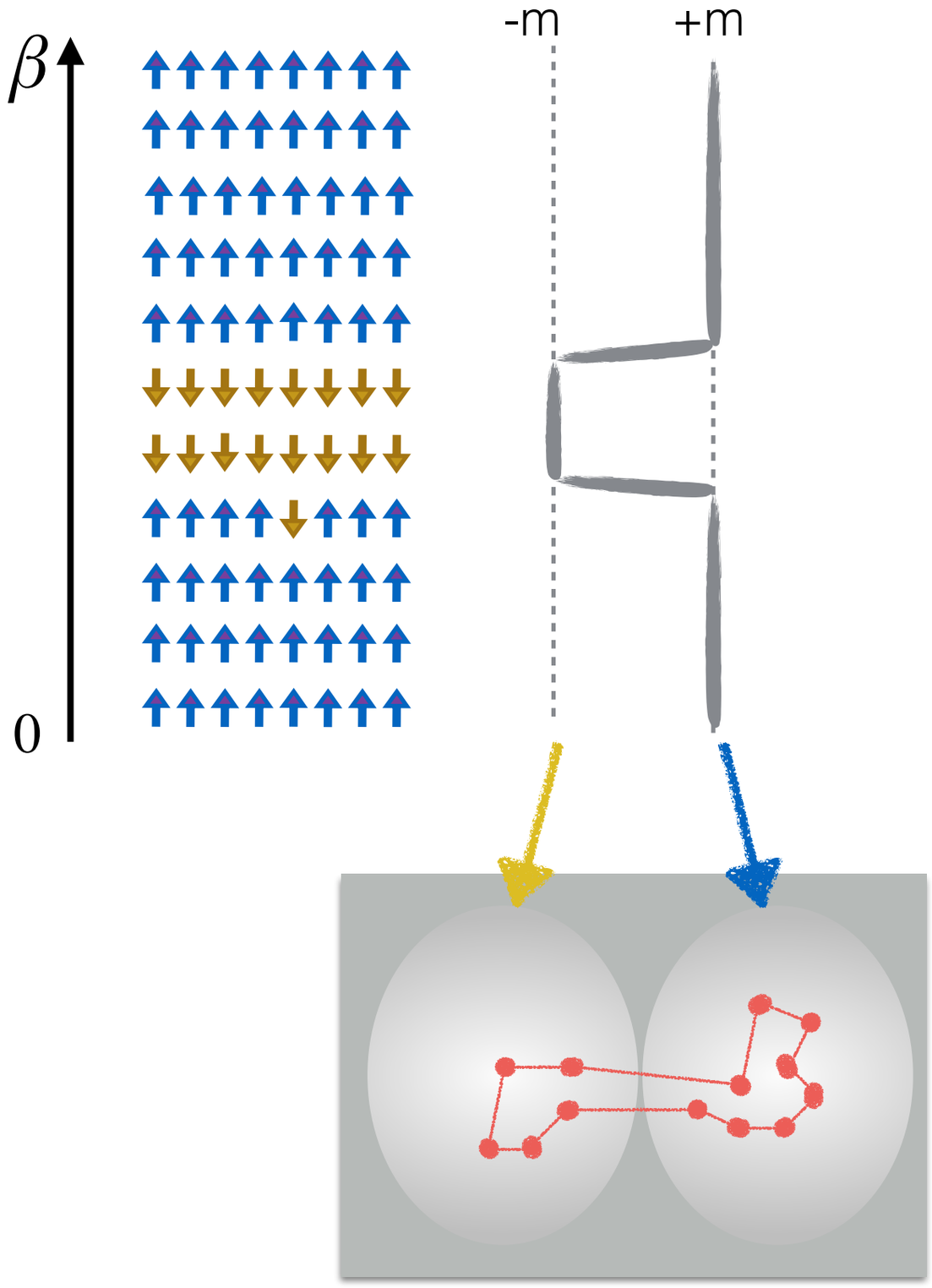


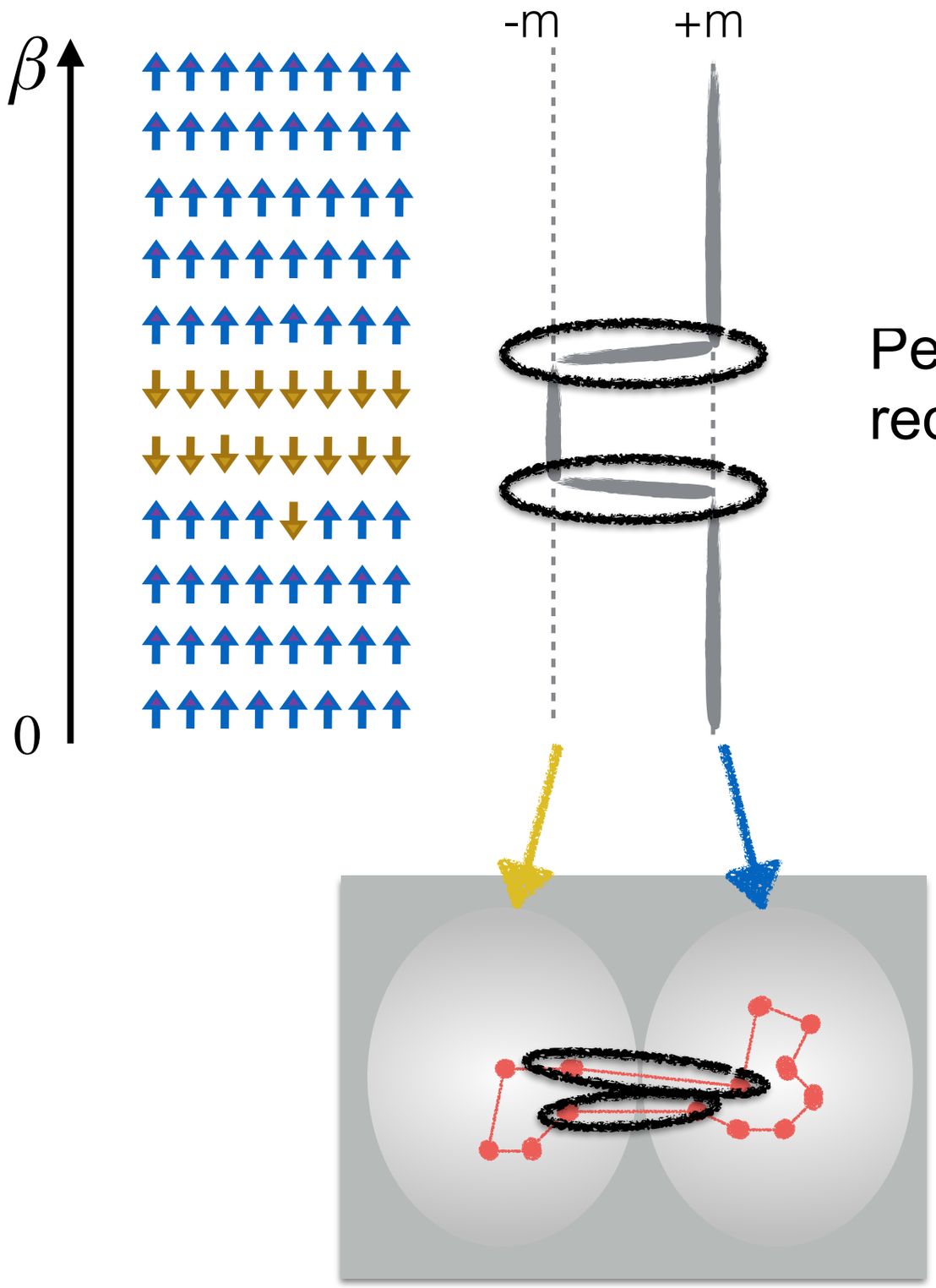
T = J/16

Tunneling rate given by QMC scale with $\sim \Delta^2$

as in physical dynamics

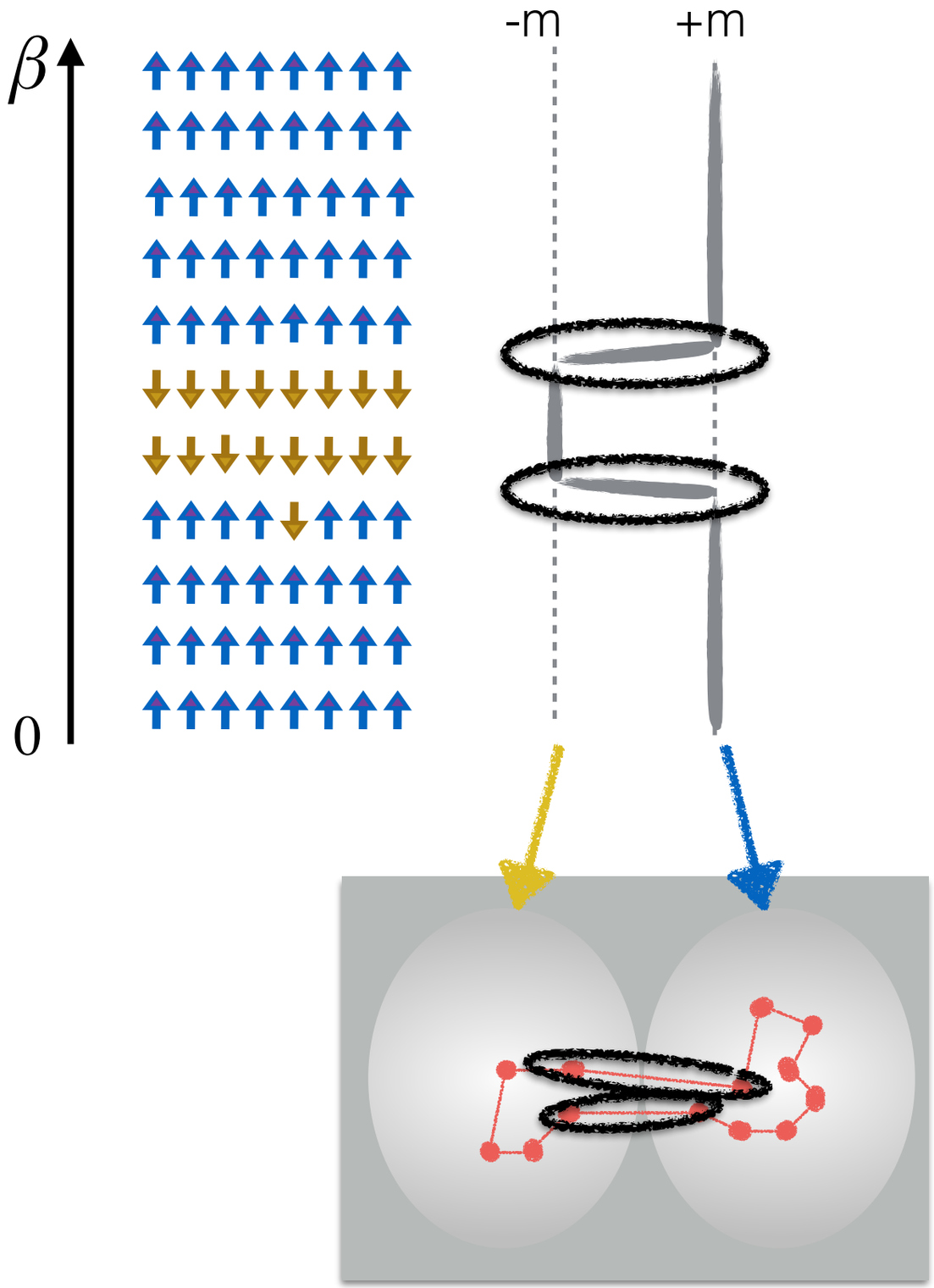

-m
+m
β
0


β
0
-m
+m

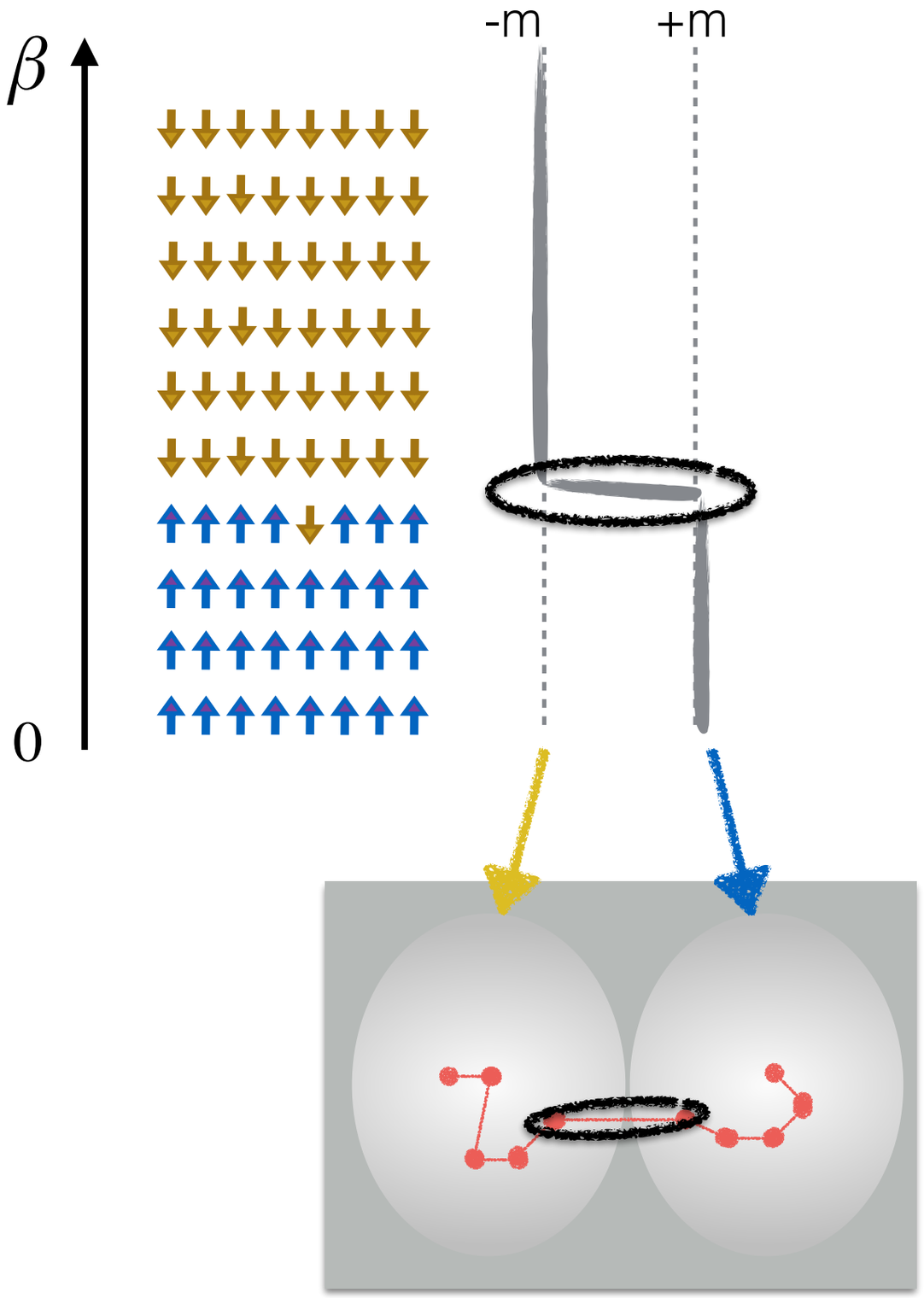


Periodic boundary conditions in imaginary time require an istanton + anti-istanton pair

$$Z = \mathrm{Tr}\, e^{-\beta H}$$



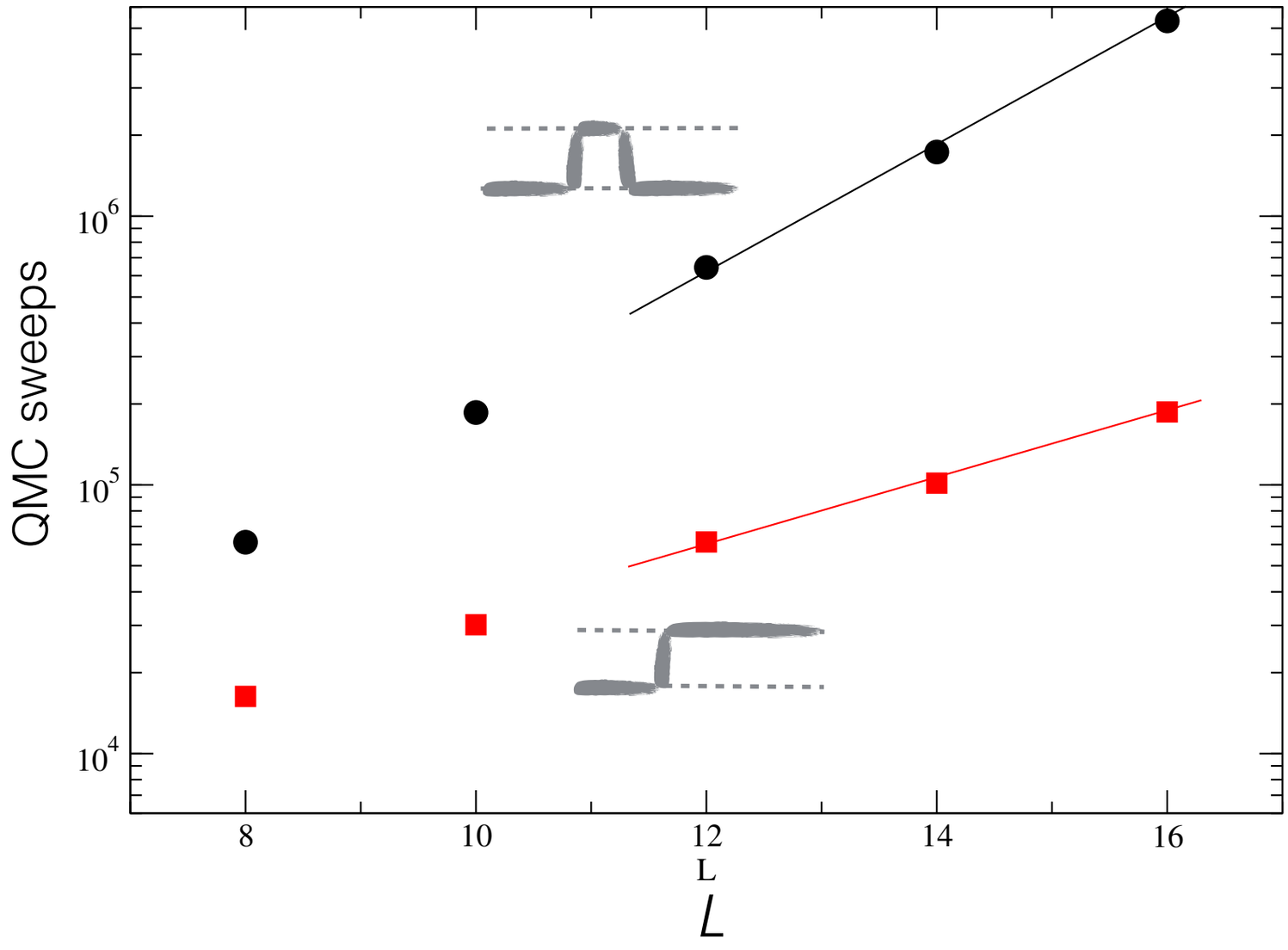
$$Z = \mathrm{Tr}\, e^{-\beta H}$$



$$K(x_0; x_M) = \langle x_0 \mid e^{-\beta H} \mid x_M \rangle$$

Scaling with system size *L*

Γ=0.8



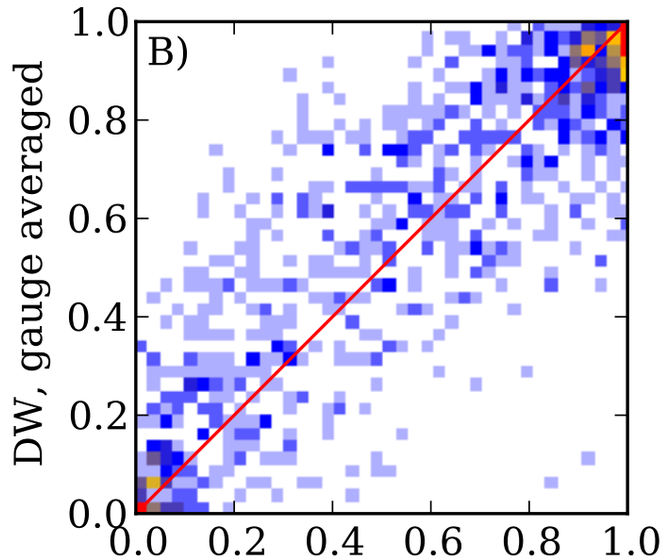
$$\tau_{PBC} \sim \frac{1}{\Delta^2}$$

$$\tau_{OBC} \sim \frac{1}{\Delta}$$

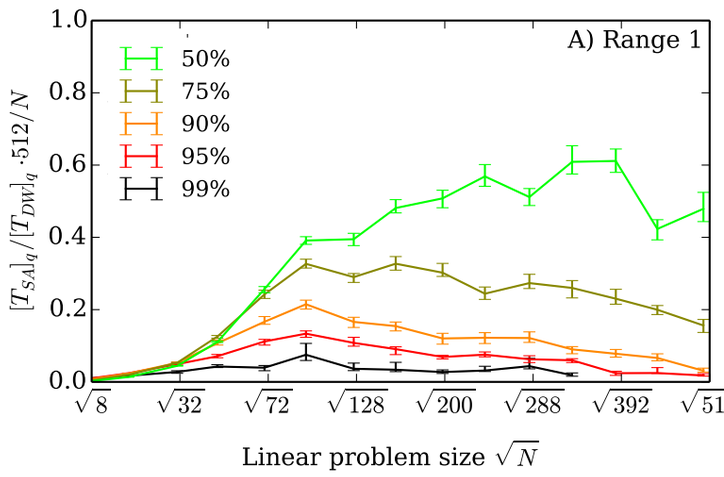
By cutting open the trace in imaginary time we have a new algorithm, with a better scaling.

We understand quantum annealing much better

- D-Wave has built a quantum annealer
  - It performs as we expect from a quantum annealer
  - but the quantum annealer may have an effective semi-classical description (Shin *et al.)*



- No evidence of quantum speedup
  - Do other classes of problems have speedup?
  - It is an analog device: is this due to calibration problems?
  - Do we need non-stoquastic quantum annealing?
  - Or is there simply no speedup in quantum annealing?



- Simulated quantum annealing
  - Scales just like true quantum annealing for tunneling through a barrier
  - Is there any advantage for a physical quantum annealer?

Collaborators

- Troels Rønnow (ETH → Nokia)
- Sergei Isakov (ETH → Google)
- Guglielmo Mazzola (ETH)
- Bettina Heim (ETH, BSc student)
- Damian Steiger (ETH)
- Ilia Zintchenko (ETH)
- Joshua Job (USC)
- Zhihui Wang (USC)
- Sergio Boixo (USC → Google)
- Dave Wecker (Microsoft)
- John Martinis (UCSB → Google)
- Daniel Lidar (USC)