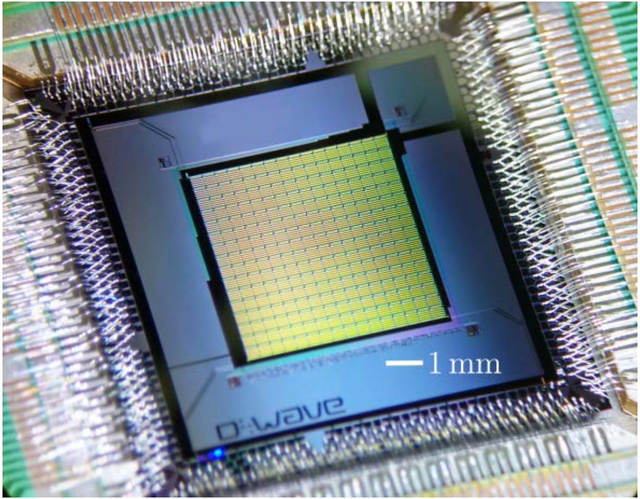
D:WAVE
The Quantum Computing Company™

A noisy brawl:
D-Wave versus D-Wave

Richard Harris

March 6, 2019

The DW2000Q quantum processing unit (QPU)



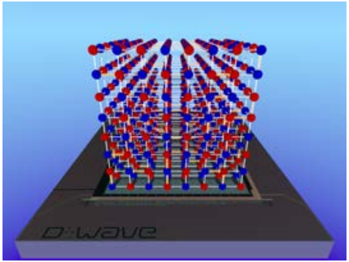
Trans.Appl.Supercond. **24**, 1700110 (2014).

We had our fun ...

Using the DW2000Q as programmable quantum matter

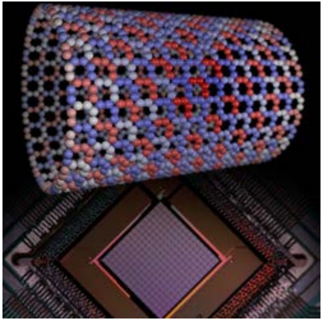
Science AAAS

nature



“Phase transitions in a programmable quantum spin glass simulator”

Science **361** 6398 162-165 (2018)



“Observation of topological phenomena in a programmable lattice of 1,800 qubits”

Nature **560** 7719 (2018)

... but where is the `quantum advantage'?

... but where is the `quantum advantage'?

(and what does coherence or noise have to do with it?)

Looking for a problem

Looking for a problem

To see the impact of noise on QA, one needs a problem with the following attributes:

Looking for a problem

To see the impact of noise on QA, one needs a problem with the following attributes:

- Favorable phase transitions.
  [See Katzgraber, Hamze, and Andrist, PRX **4**, 021008 (2014)]

Looking for a problem

To see the impact of noise on QA, one needs a problem with the following attributes:

- Favorable phase transitions.
  [See Katzgraber, Hamze, and Andrist, PRX **4**, 021008 (2014)]
- Hard enough to resolve optimal anneal time.
  [See Albash and Lidar, PRX **8**, 031016 (2018)]

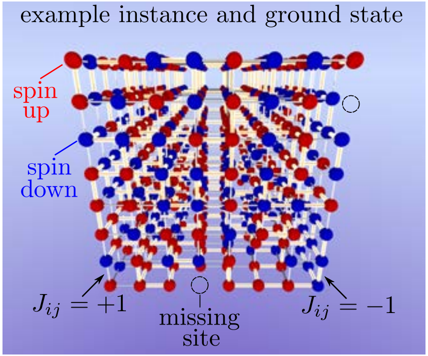
Looking for a problem

To see the impact of noise on QA, one needs a problem with the following attributes:

- Favorable phase transitions.
  [See Katzgraber, Hamze, and Andrist, PRX **4**, 021008 (2014)]
- Hard enough to resolve optimal anneal time.
  [See Albash and Lidar, PRX **8**, 031016 (2018)]
- Ground state probability is sensitive to noise within experimental constraints.
  [?]

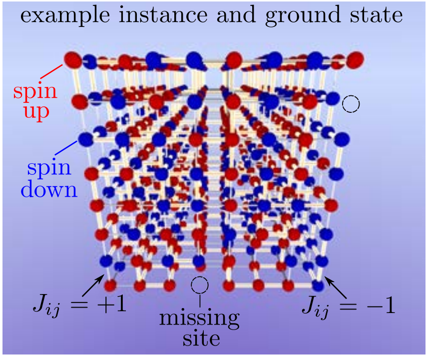
$8 \times 8 \times 8$ cubic lattice spin glasses

An embedded problem that has the desired attributes. [See *Science* **361**, 162 (2018)]



$8 \times 8 \times 8$ cubic lattice spin glasses

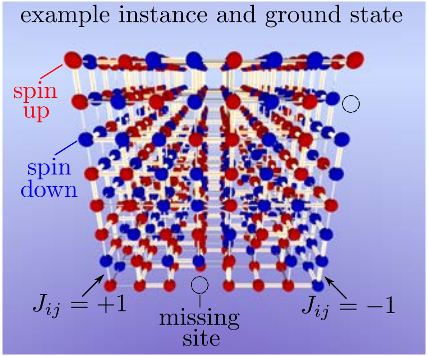
An embedded problem that has the desired attributes. [See *Science* **361**, 162 (2018)]



- Favorable *phase transitions* as a function of quantum annealing parameter s. ✓

$8 \times 8 \times 8$ cubic lattice spin glasses

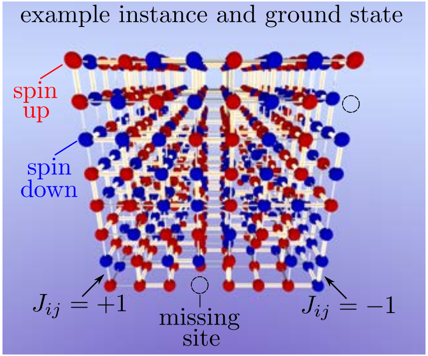
An embedded problem that has the desired attributes. [See *Science* **361**, 162 (2018)]



- Favorable *phase transitions* as a function of quantum annealing parameter s. ✓
- Hard enough to make a DW2000Q QPU sweat. ✓

$8 \times 8 \times 8$ cubic lattice spin glasses

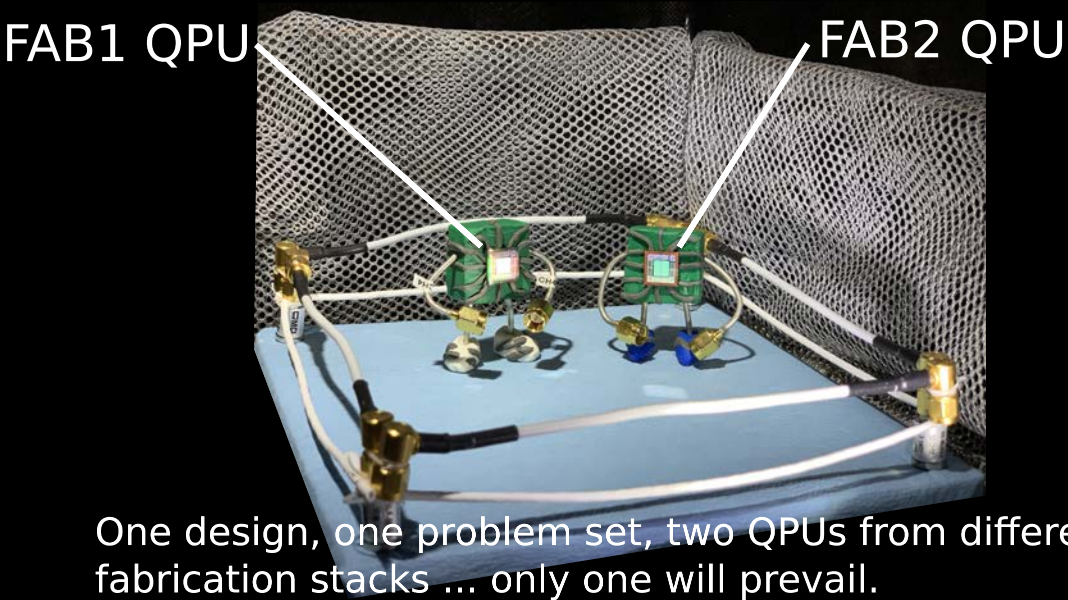
An embedded problem that has the desired attributes. [See *Science* **361**, 162 (2018)]



- Favorable *phase transitions* as a function of quantum annealing parameter s. ✓
- Hard enough to make a DW2000Q QPU sweat. ✓
- **. . . but are these problems sensitive to noise?**

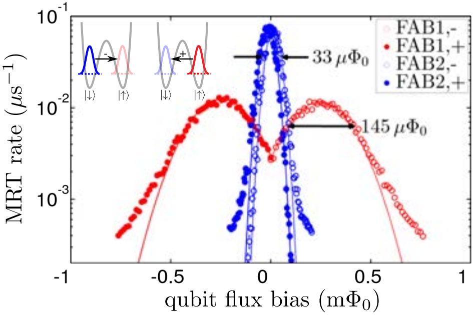
Cage match

The contenders



One design, one problem set, two QPUs from different fabrication stacks ... only one will prevail.

Two identical QPUs but with different noise characteristics



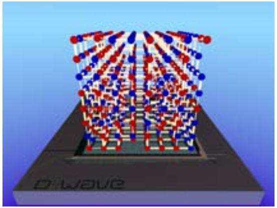
- Two QPUs manufactured with different processes:
  - **FAB1** - variant of process used to manufacture DW2000Q products
  - **FAB2** - a more recent experimental fabrication stack
- Roughly a factor of 5× less noise in FAB2 relative to FAB1.

The rules

Solving cubic lattice spin glasses

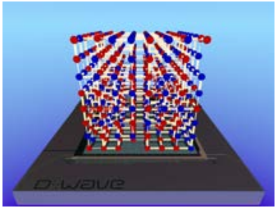
Solving cubic lattice spin glasses

1. Send a spin glass instance to the QPU, take lots (10^5) of reads for a range of anneal time t_a.

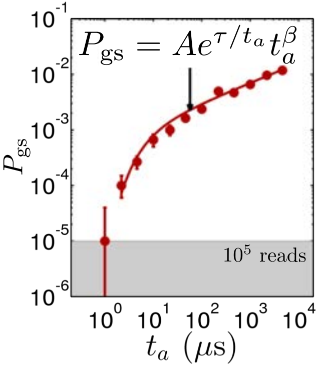


Solving cubic lattice spin glasses

1. Send a spin glass instance to the QPU, take lots (10^5) of reads for a range of anneal time t_a.

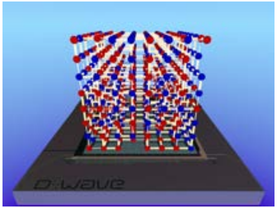


2. Record the probability of observing a ground state P_{gs} .

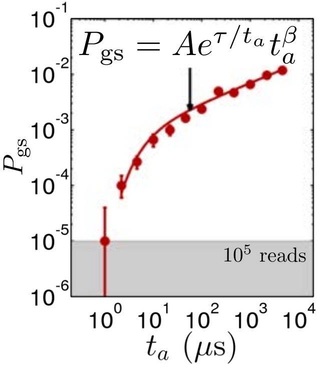


Solving cubic lattice spin glasses

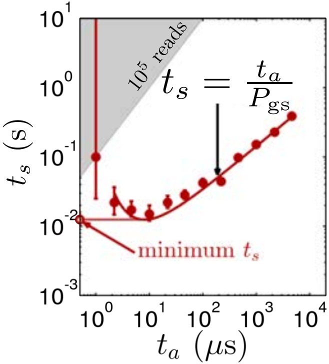
1. Send a spin glass instance to the QPU, take lots (10^5) of reads for a range of anneal time t_a.



2. Record the probability of observing a ground state P_{gs} .

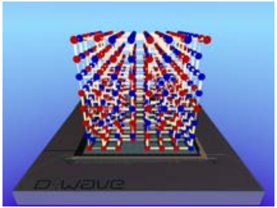


3. Convert to solution time t_s . Determine optimal t_a and minimum t_s

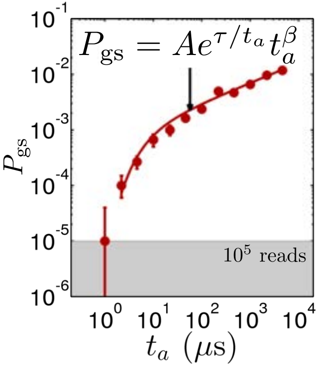


Solving cubic lattice spin glasses

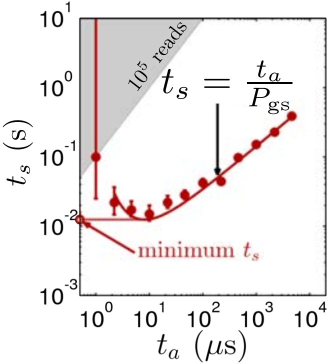
1. Send a spin glass instance to the QPU, take lots (10^5) of reads for a range of anneal time t_a.



2. Record the probability of observing a ground state P_{gs} .



3. Convert to solution time t_s . Determine optimal t_a and minimum t_s

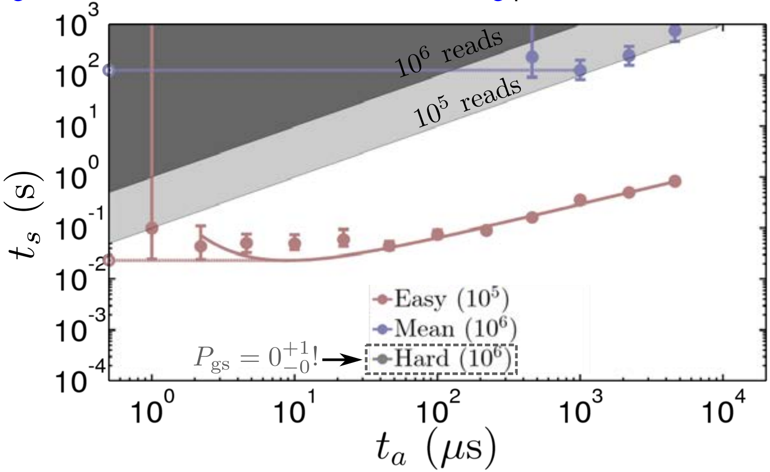


4. Repeat steps 1-3 for 100 randomly generated spin glass instances on both FAB1 and FAB2 QPUs. Note that identical instances are run on each QPU. ***Which QPU wins?***

Warm-up: Getting the embedding right

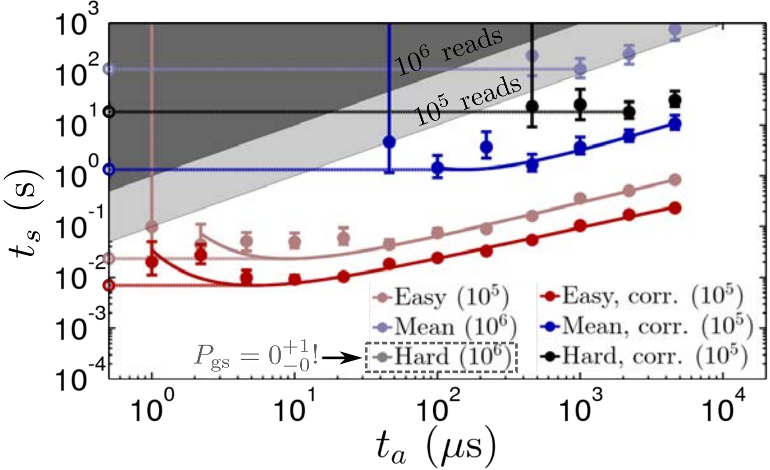
Correcting the embedding improves performance

Consider 3 example instances run on FAB1 QPU. Measure t_s versus t_a for a naive embedding and then a corrected (corr.) embedding per *Science* **361** 6398 162-165 (2018):



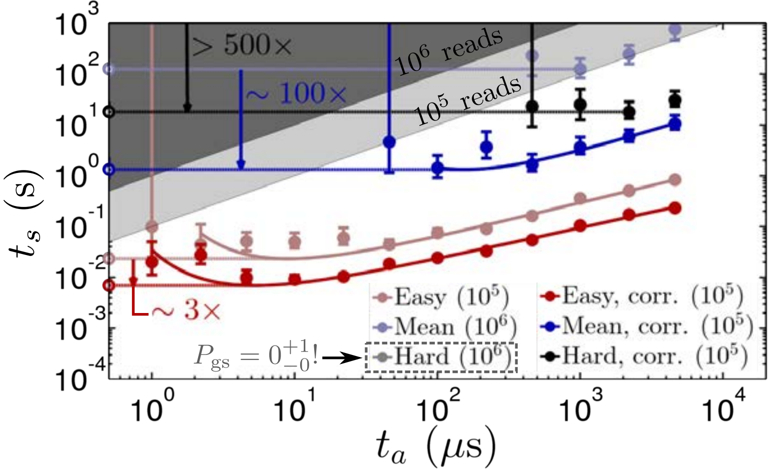
Correcting the embedding improves performance

Consider 3 example instances run on FAB1 QPU. Measure t_s versus t_a for a naive embedding and then a corrected (corr.) embedding per *Science* **361** 6398 162-165 (2018):



Correcting the embedding improves performance

Consider 3 example instances run on FAB1 QPU. Measure t_s versus t_a for a naive embedding and then a corrected (corr.) embedding per *Science* **361** 6398 162-165 (2018):

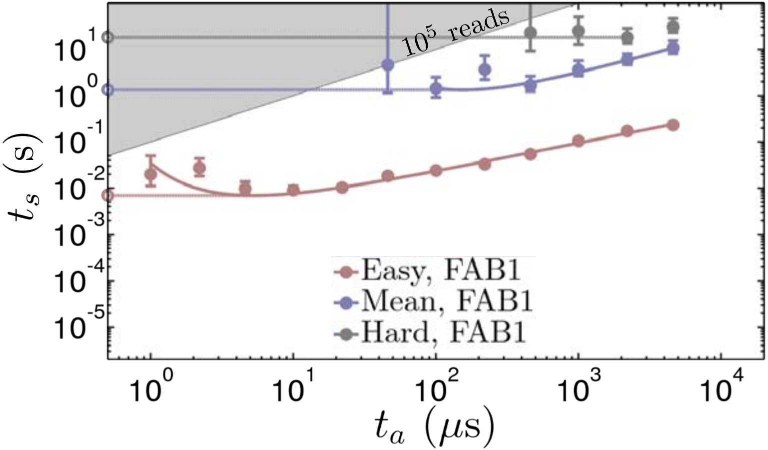


- Correcting → $>500\times$ reduction in optimal solution time t_s on hard instance.

The main event

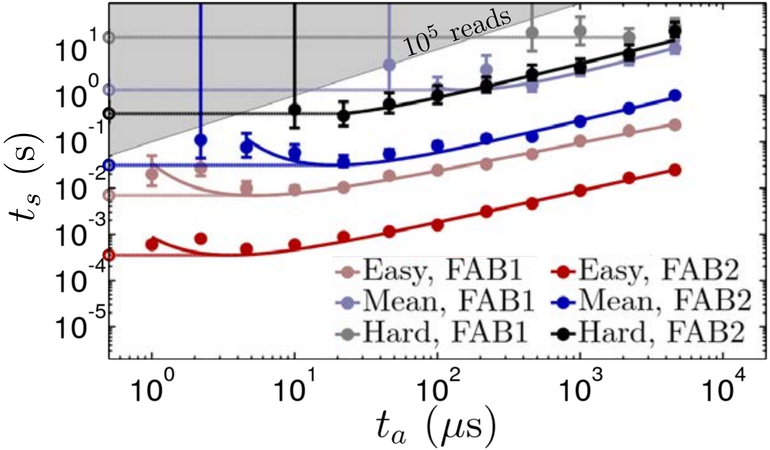
Reducing noise improves performance

Consider the same 3 example instances run on both FAB1 and FAB2 QPUs with corrections per *Science* **361** 6398 162-165 (2018):



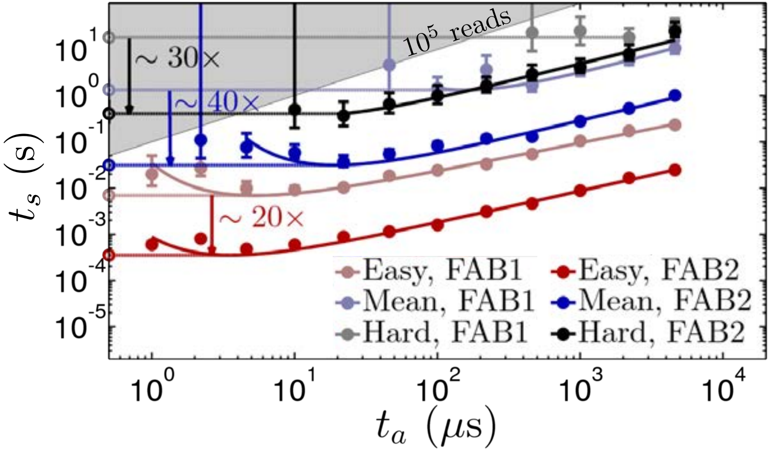
Reducing noise improves performance

Consider the same 3 example instances run on both FAB1 and FAB2 QPUs with corrections per *Science* **361** 6398 162-165 (2018):



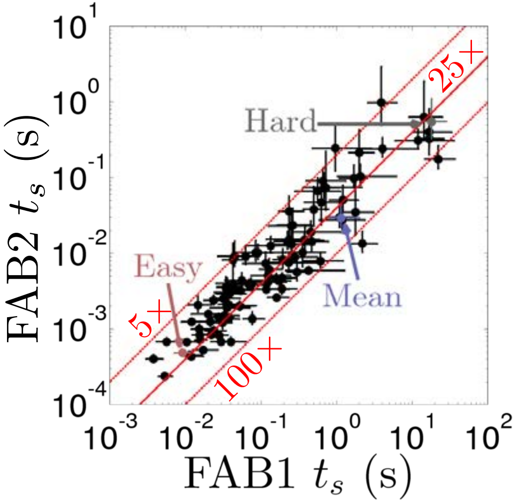
Reducing noise improves performance

Consider the same 3 example instances run on both FAB1 and FAB2 QPUs with corrections per *Science* **361** 6398 162-165 (2018):



- Modest 5× reduction in noise → up to 40× reduction in optimal solution time t_s.

All 100 instances



- Mean 25× reduction in optimal solution time t_s.

Conclusions

Conclusions

Conclusions

- Use the lessons learned from studying phase transitions in embedded problems to condition the embedding ($\sim 100\times$ improvement on typical instances, $> 500\times$ on the hardest instances).

Conclusions

- Use the lessons learned from studying phase transitions in embedded problems to condition the embedding ($\sim 100\times$ improvement on typical instances, $> 500\times$ on the hardest instances).
- Modest reductions in noise give significant reductions in solution time ($\sim 5\times$ reduction in noise, $\sim 25\times$ improvement on typical instances).

Conclusions

- Use the lessons learned from studying phase transitions in embedded problems to condition the embedding ($\sim 100\times$ improvement on typical instances, $> 500\times$ on the hardest instances).
- Modest reductions in noise give significant reductions in solution time ($\sim 5\times$ reduction in noise, $\sim 25\times$ improvement on typical instances).
- Further reductions in noise will continue to improve performance. We are nowhere near a fundamental limit.