Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

Introduction to Quantum Machine Learning

Gabriel Perdue // @Fermilab

Fermilab 2021 Summer Student School at LNF

2021 / August / 4

Special thanks to Q. Zhuang (U. of Arizona) for a recent review seminar I draw from here.

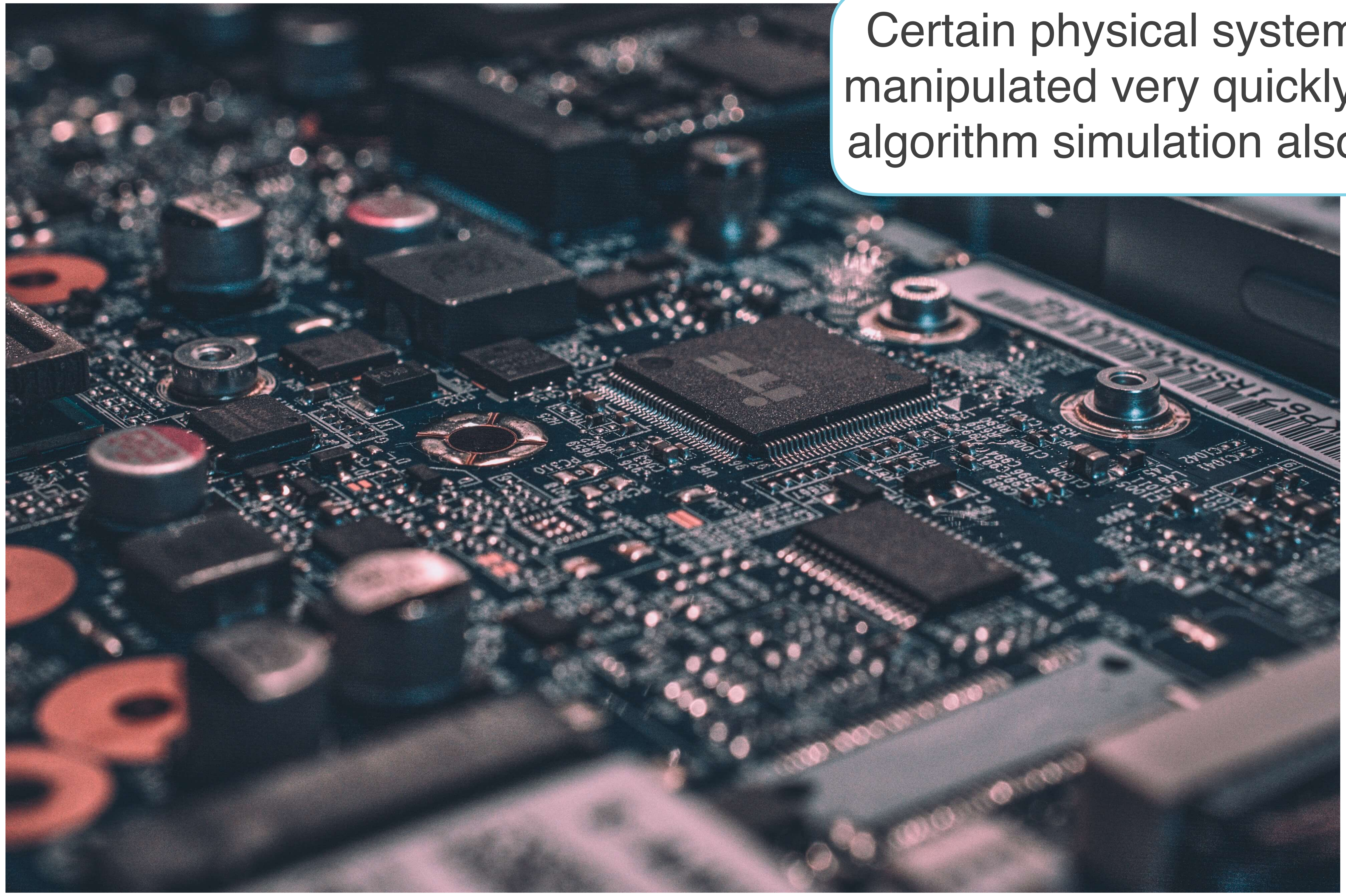
Overview

- Computing… what is it? (An opinion.)
- Quantum computing… what is it? (Another opinion.)
- Flavors of quantum machine learning
  - Classical data on quantum machines
  - Classical ML for quantum machines
  - Quantum data on quantum machines

Note: I am not discussing machine learning (ML) in an introductory or pedagogical way. I assume the audience knows the basic concepts of ML, and has some familiarity with famous algorithms like Support Vector Machines (SVMs), decision trees, and neural networks. Expert knowledge of these topics is not required, but everyone hopefully understands the idea of heuristic algorithms that may be tuned by (or fit to) data.

What is computing?

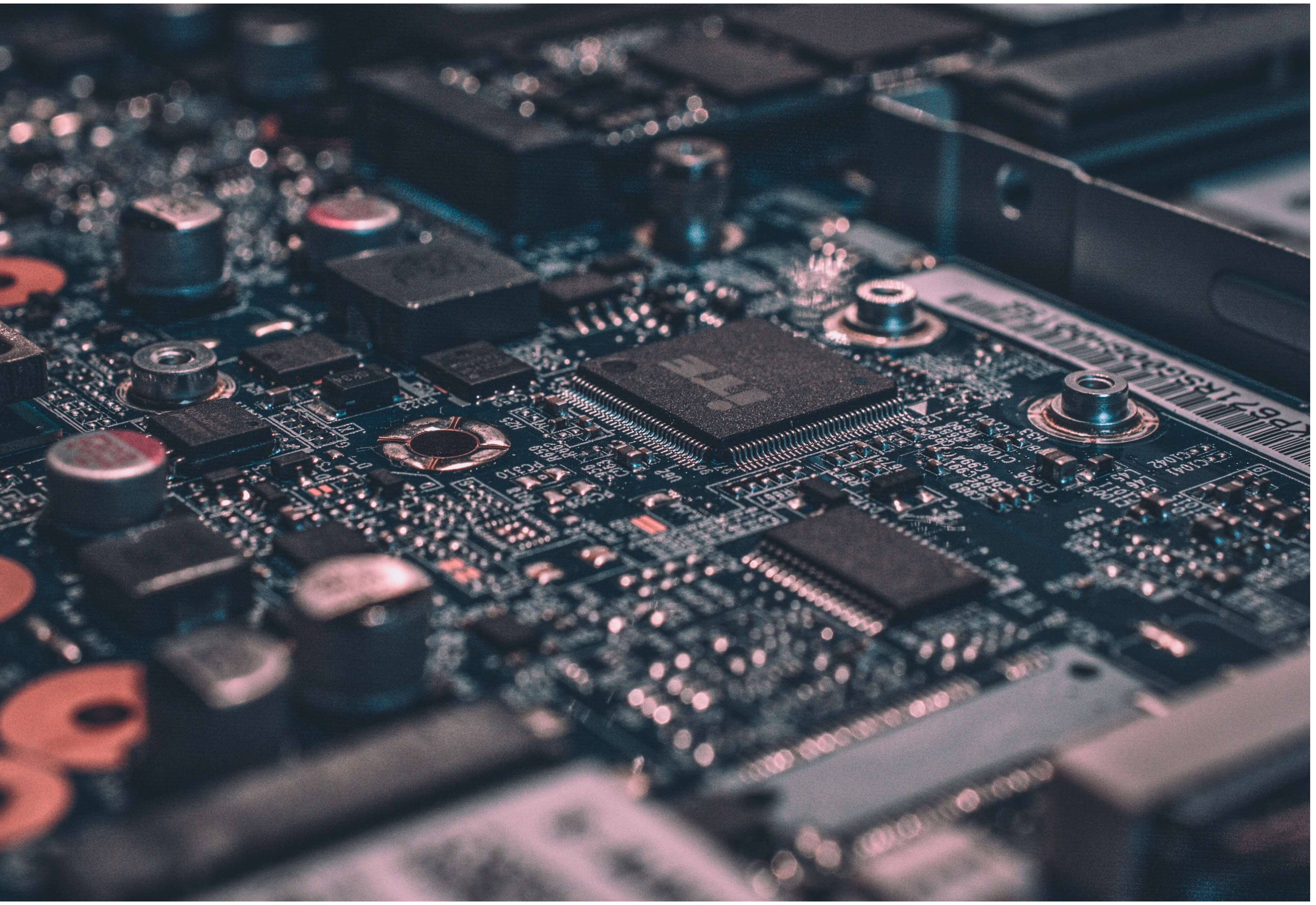
- First, what is computing? One perspective - it is ***physical simulation of algorithms coupled to interpretation***. We manipulate a physical system according to rules. A metaphysical tower of concepts then allows us to *interpret* the results.

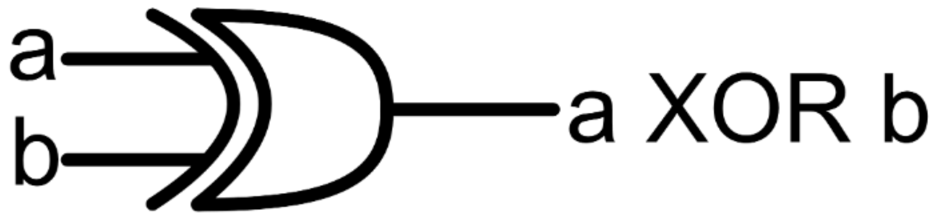


Certain physical systems can be manipulated very quickly - making algorithm simulation also very fast.

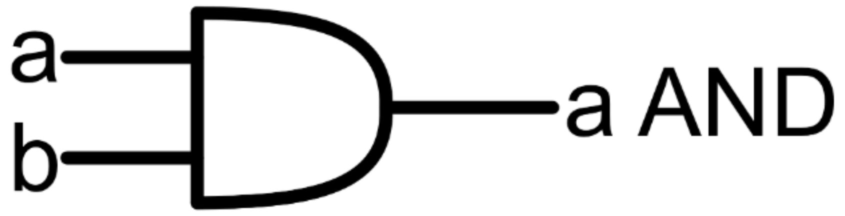
We can simulate algorithms blindly - ultimately *interpretation* is required.

What is classical computing?





	b = 0	b = 1
a = 0	0	1
a = 1	1	0

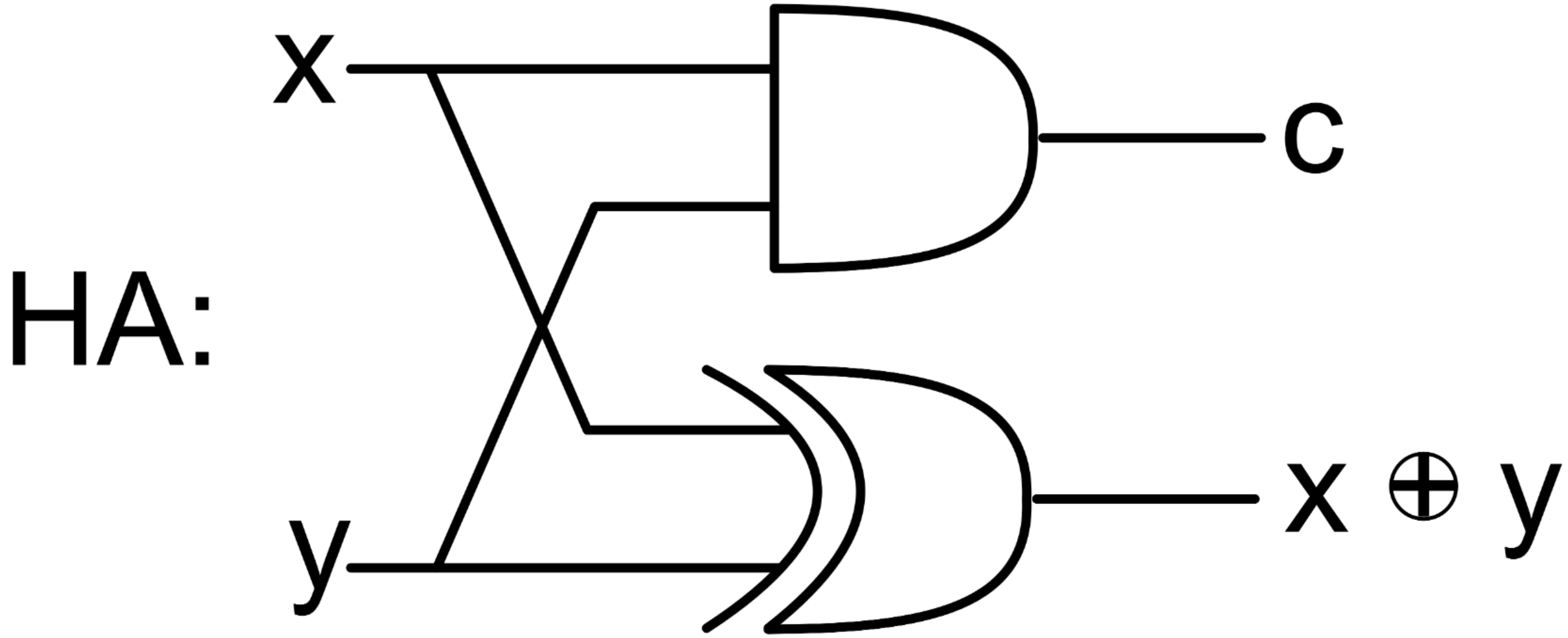


	b = 0	b = 1
a = 0	0	0
a = 1	0	1

You can see how you would implement a table like this one with logic gates:

```
  0     0     1     1
 +0    +1    +0    +1
---   ---   ---   ---
 00    01    01    10
```

You need two inputs and two outputs. This function is called a *Half Adder*:

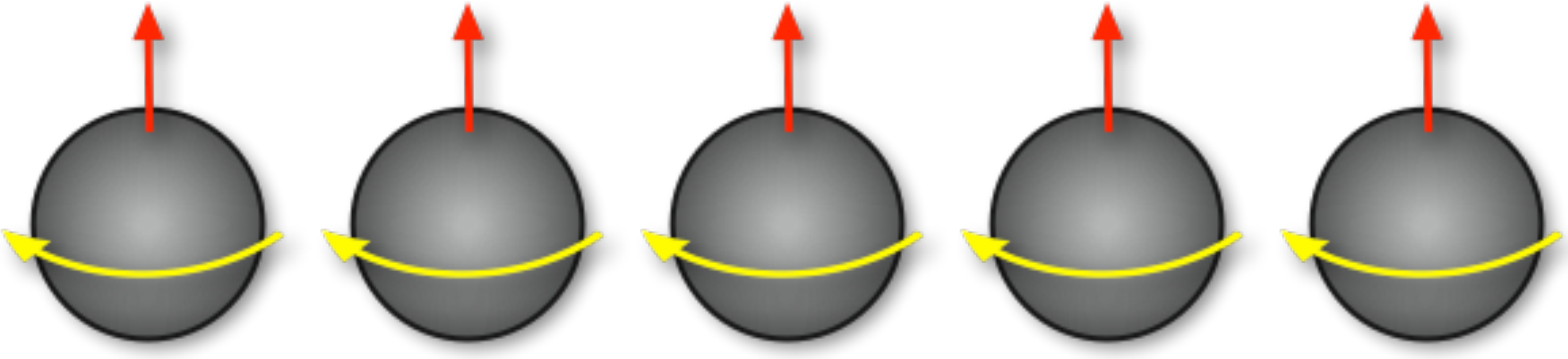


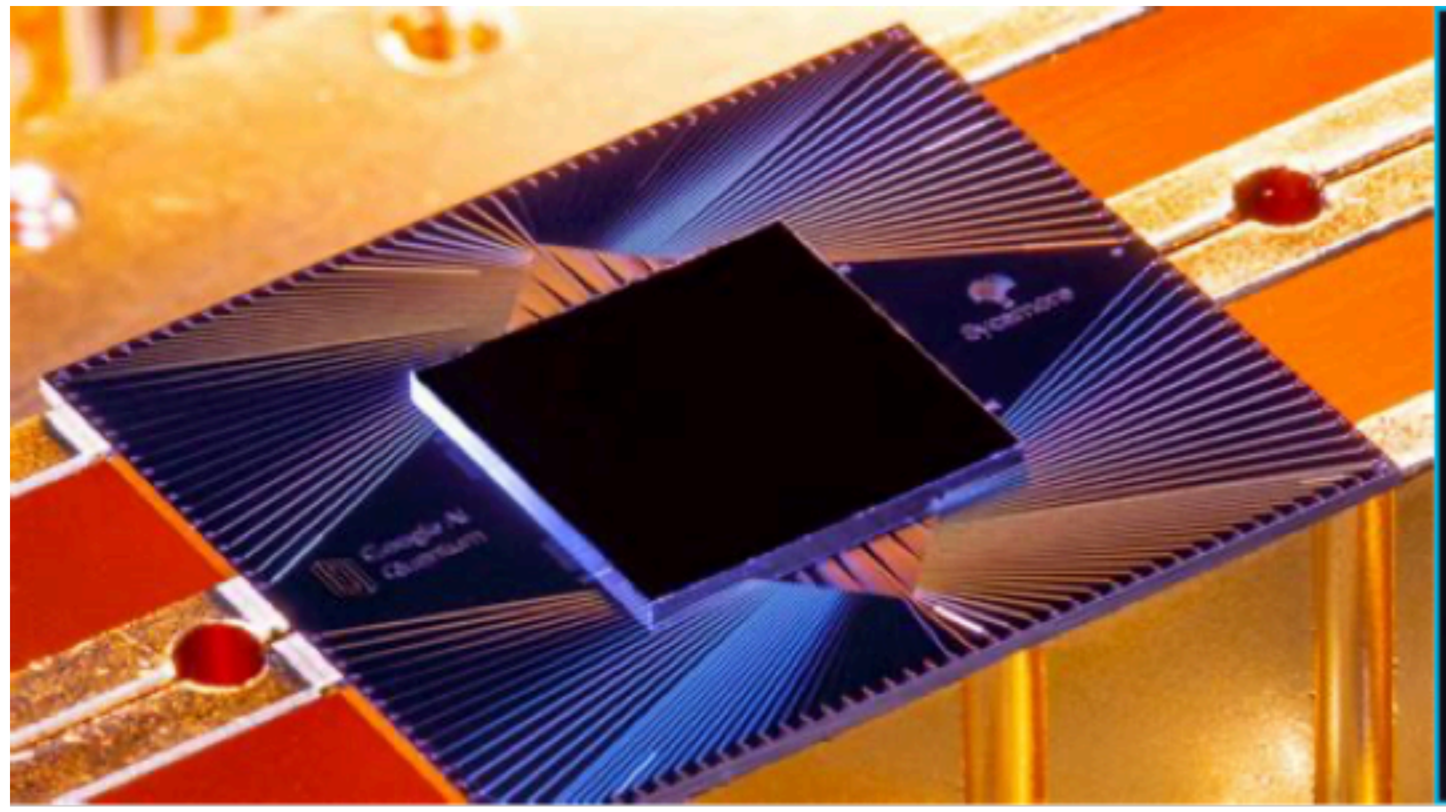
0	0	1	1	x
+0	+1	+0	+1	+y
00	01	01	10	$c(x \oplus y)$

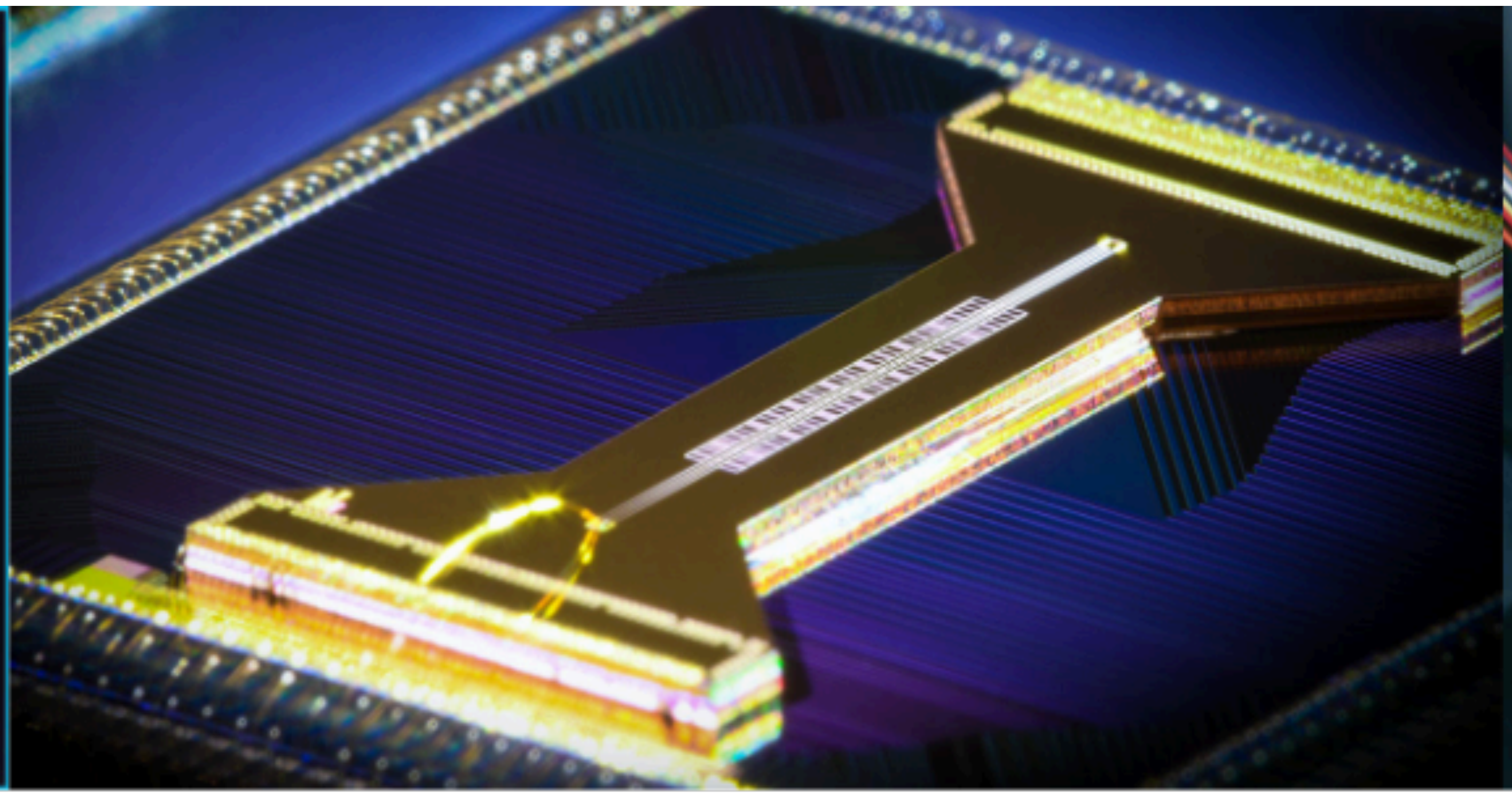
What is *quantum* computing?

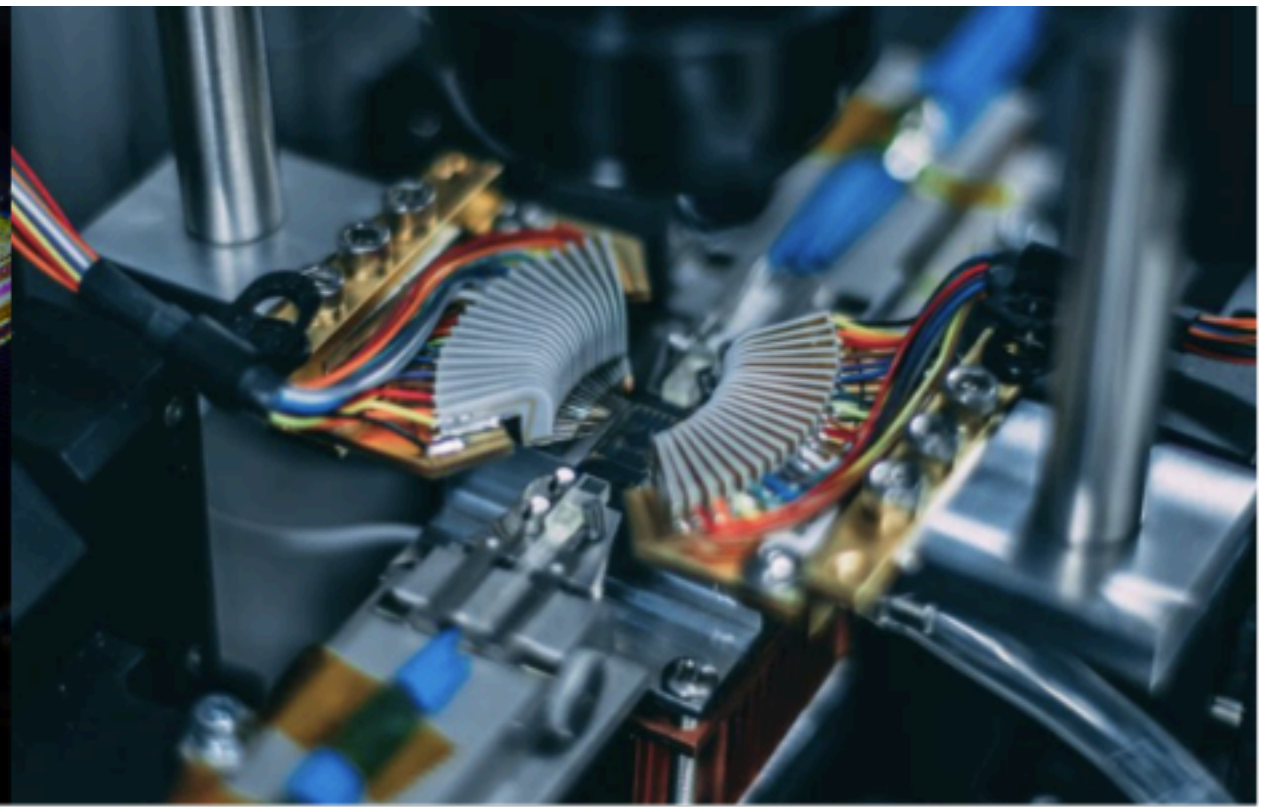
- Quantum computing is using quantum systems to simulate our algorithms.
- Challenges are rooted in the fact that quantum systems are *delicate*. And algorithms are non-obvious.
- Multiple, “competing” platforms for quantum computation exist. The ultimate goals are *scale* and *quantum error correction*.

There are many ways to leverage quantum systems to simulate an algorithm. Features of quantum measurement mean the calculations are probabilistic.









https://ai.googleblog.com/2019/10/quantum-supremacy-using-programmable.html
https://sqms.fnal.gov/research/

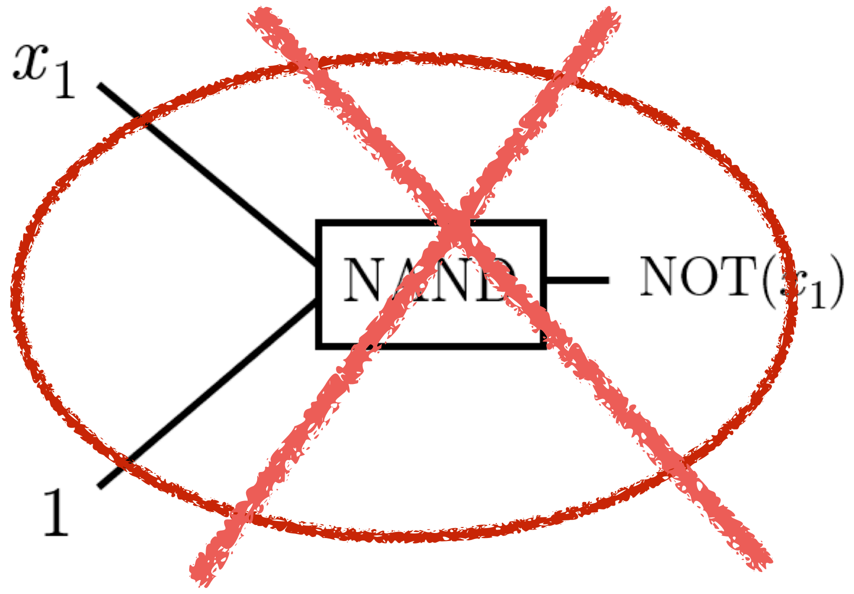
https://www.honeywell.com/en-us/company/quantum
https://www.xanadu.ai/hardware

What *is* quantum computing?

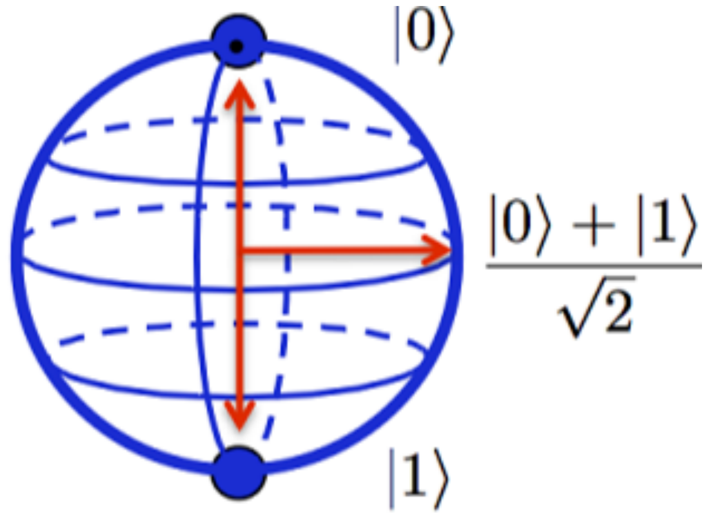
- At heart, quantum computing is unitary evolution of quantum states.
- It is distinguished by the following features:
  - Entanglement
  - Unitary evolution
  - Superposition of states
  - Reversible computation
  - Probabilistic computation
  - Exponential Hilbert spaces
  - Challenges with state coherence

Quantum computing power* scales exponentially with qubits
N bits can exactly simulate log N qubits

This compute unit....			
	Commodore 64	AWS M4 Instance	Entire Global Cloud
		1 Million x Commodore 64	1 Billion x (1 Million x Commodore 64)
can exactly simulate:	**10 Qubits**	**30 Qubits**	**60 Qubits**



Classical Bit

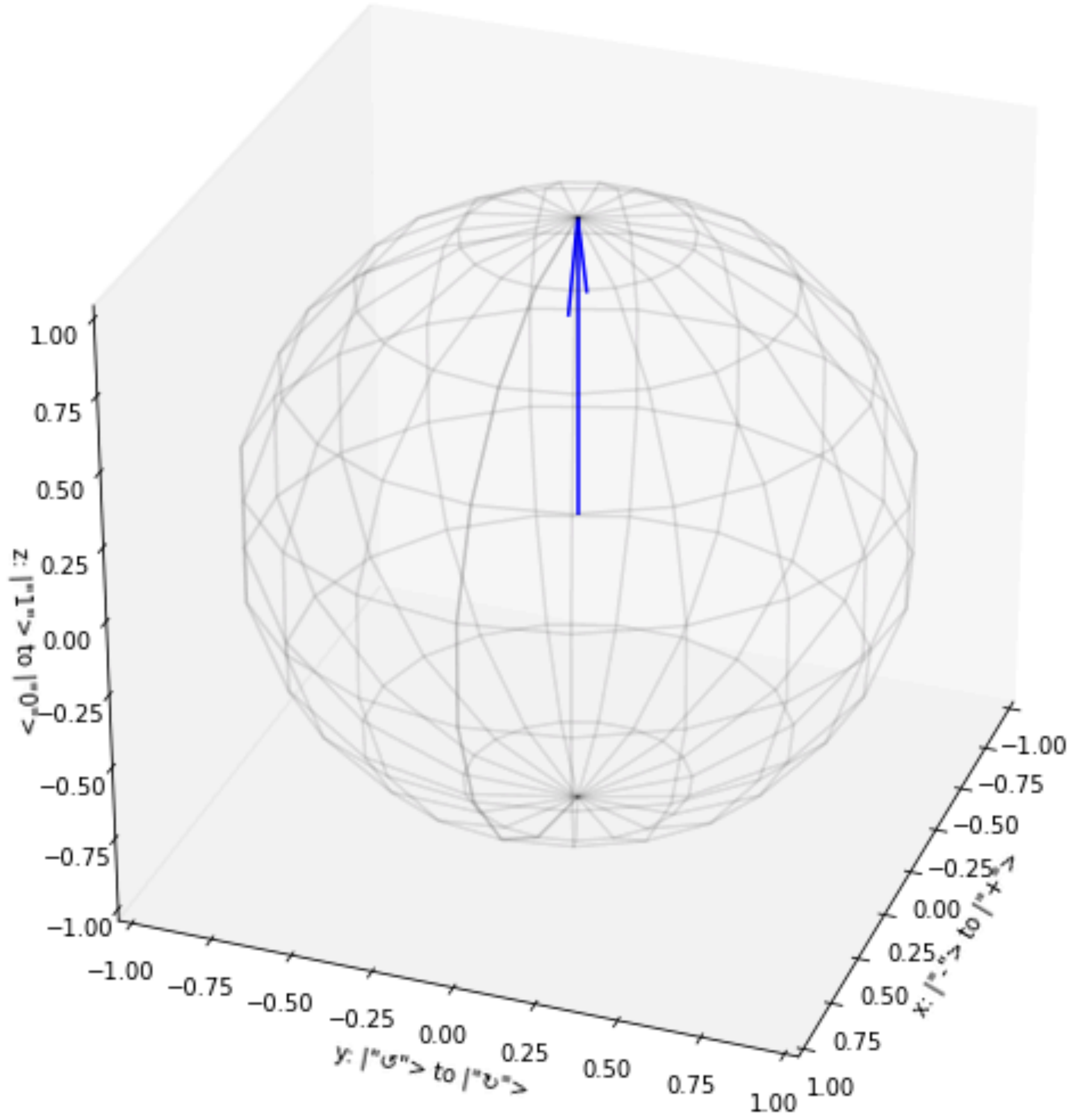


Qubit

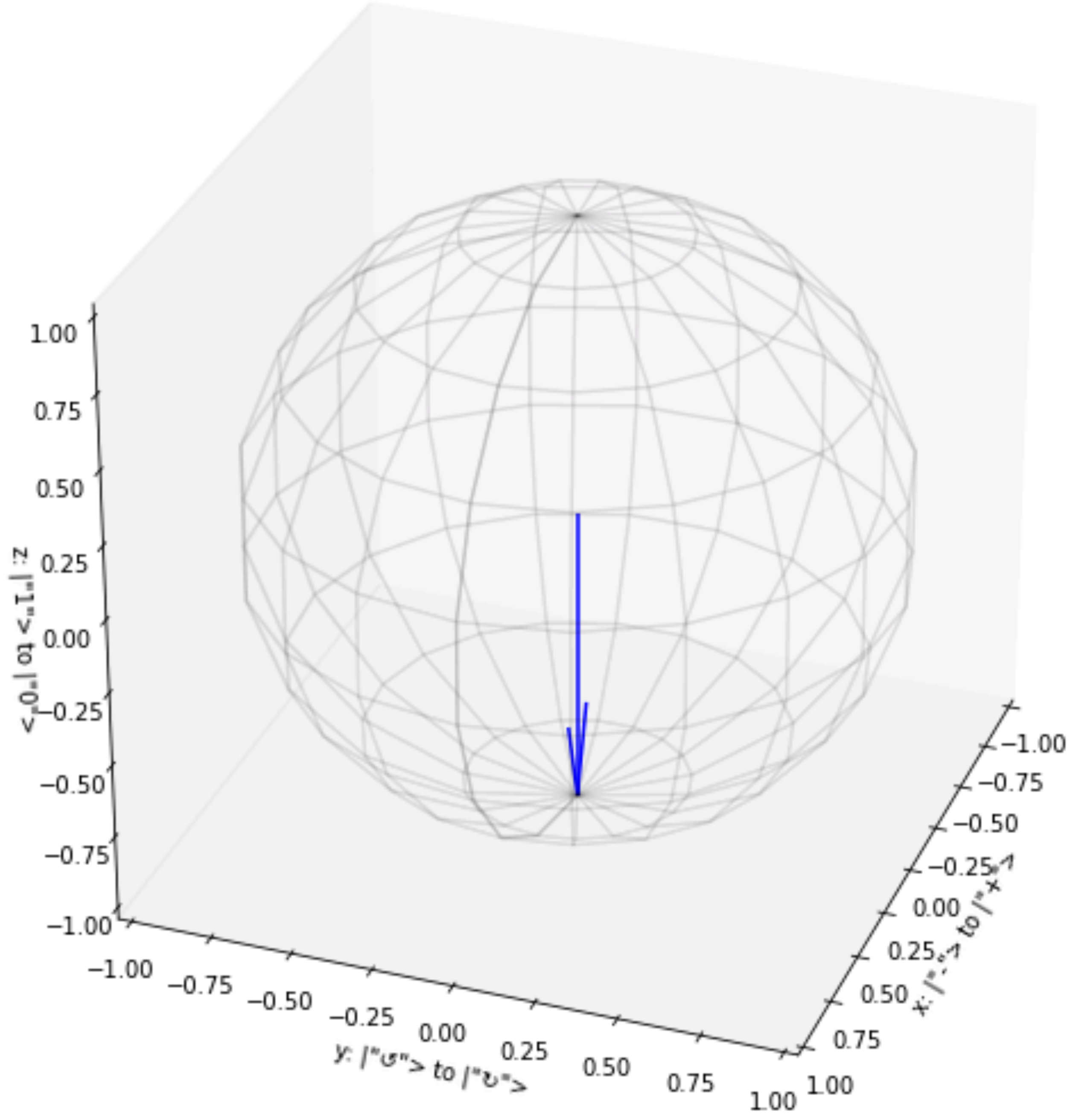
Qubits

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \alpha \times \begin{pmatrix}1\\ 0\end{pmatrix} + \beta \times \begin{pmatrix}0\\ 1\end{pmatrix} \equiv \alpha|0\rangle + \beta|1\rangle$$

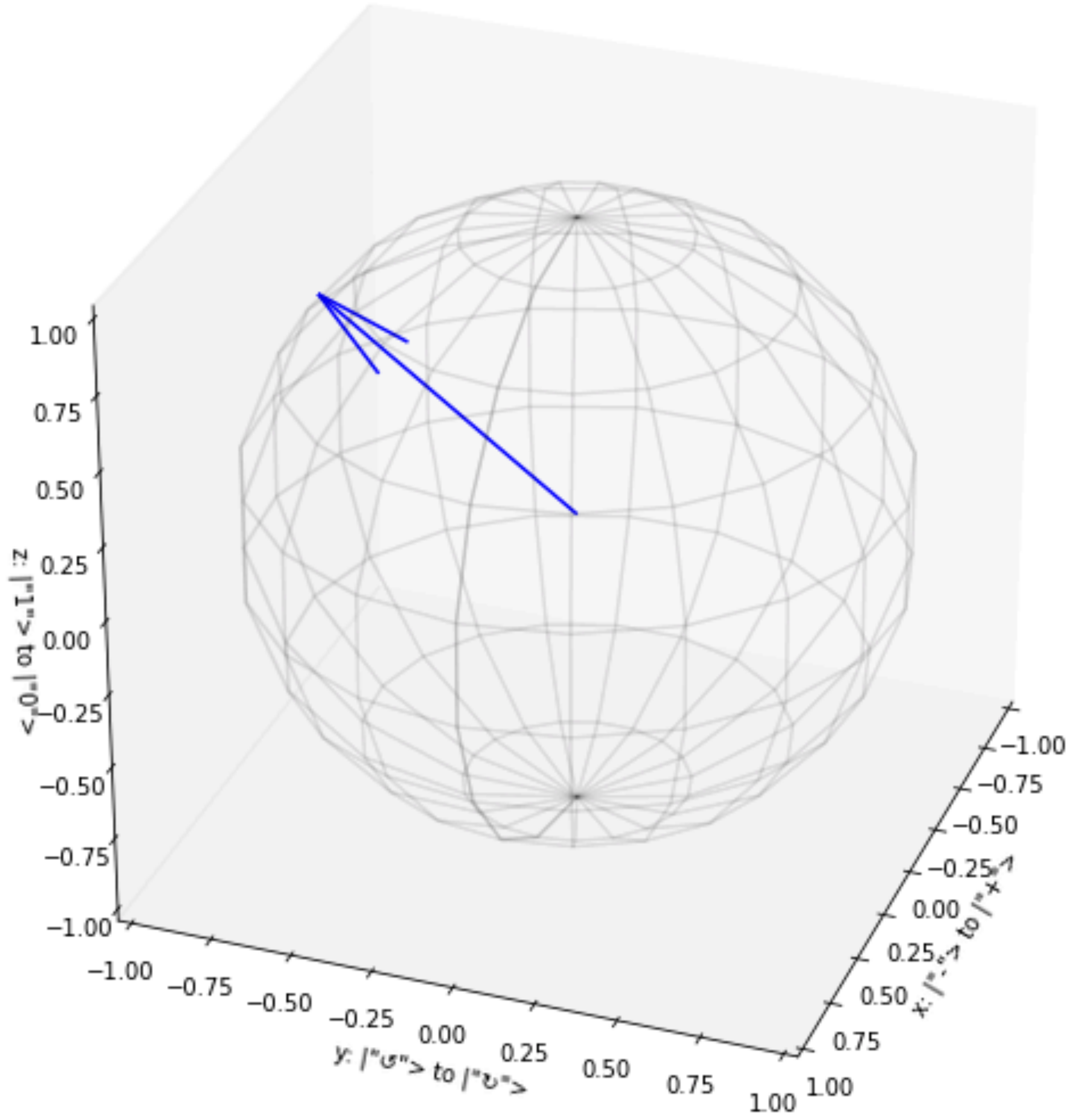
Quantum operators rotate the vector’s direction.



$|0\rangle$



$|1\rangle$



$\sqrt{0.2}|0\rangle + \sqrt{0.8}|1\rangle$

Fermilab

What *is* quantum computing?

https://bit.ly/38bidph

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \alpha \times \begin{pmatrix}1\\0\end{pmatrix} + \beta \times \begin{pmatrix}0\\1\end{pmatrix} \equiv \alpha|0\rangle + \beta|1\rangle$$

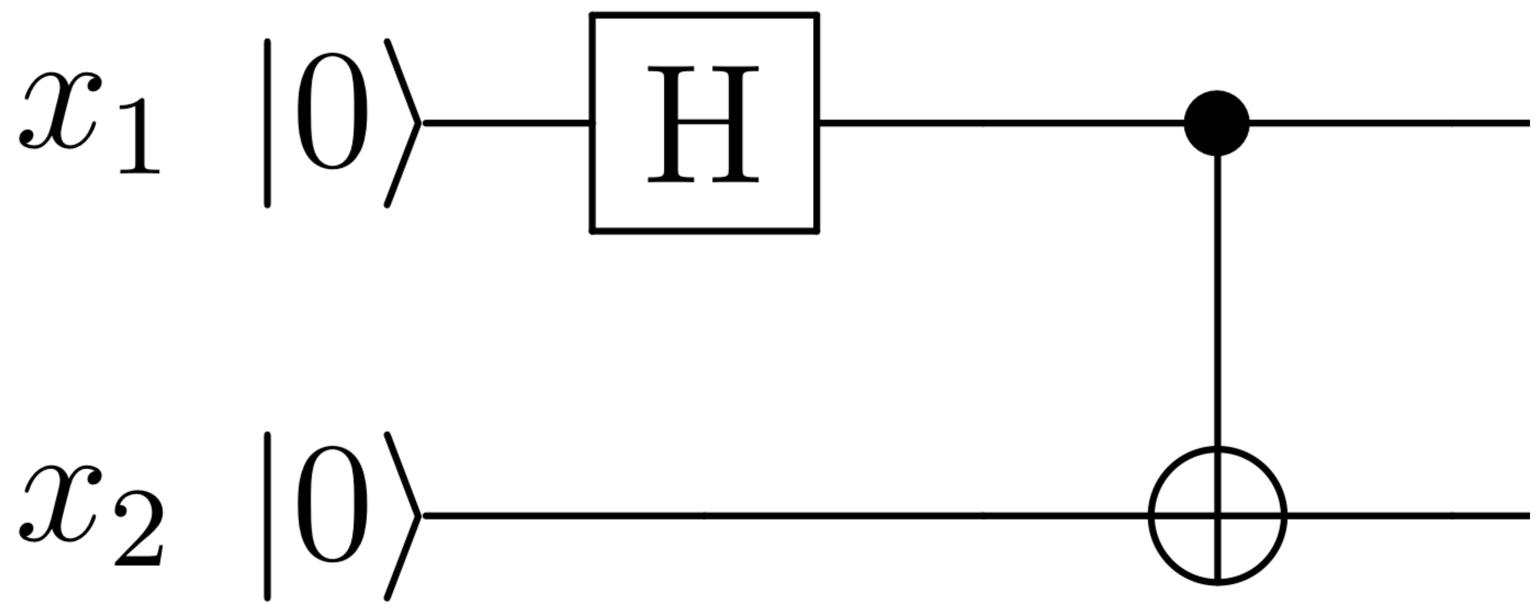
$$|0\rangle|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\begin{pmatrix}1\\0\end{pmatrix}\\ 0\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} = |00\rangle$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \qquad X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$$

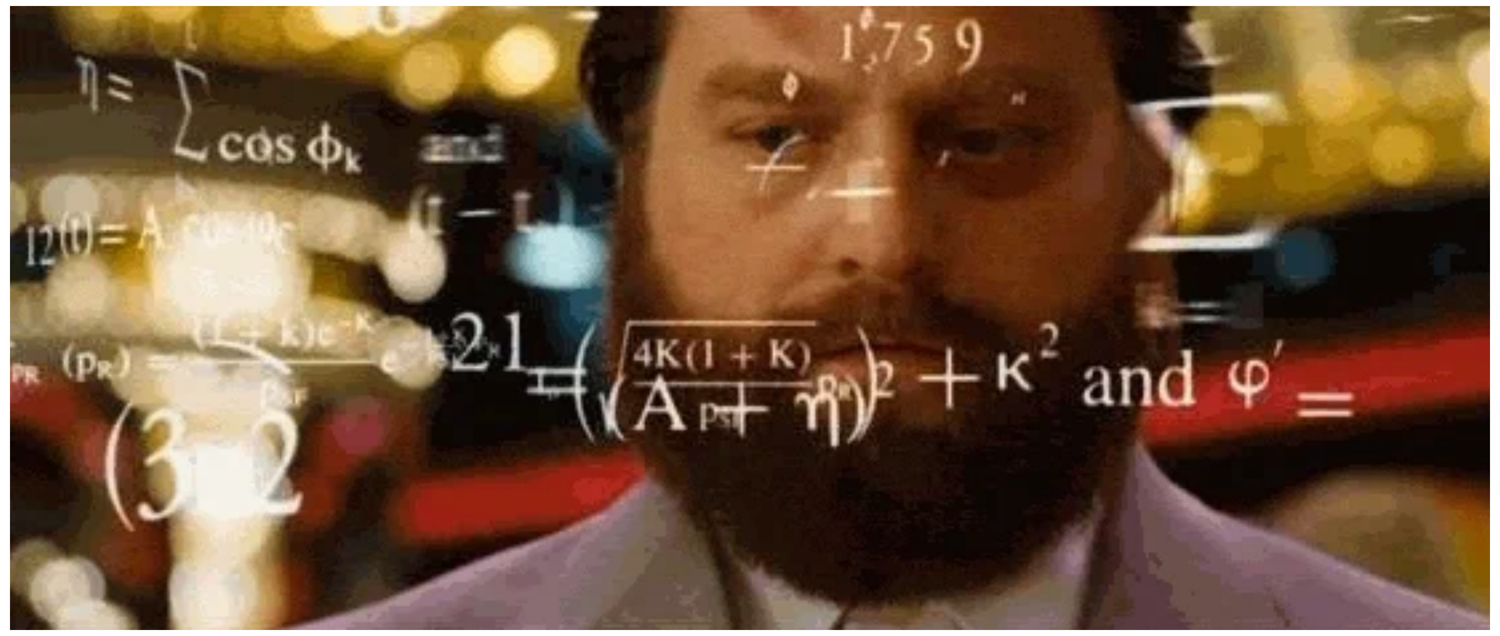
$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle$$

Superposition!



$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{pmatrix} \quad \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

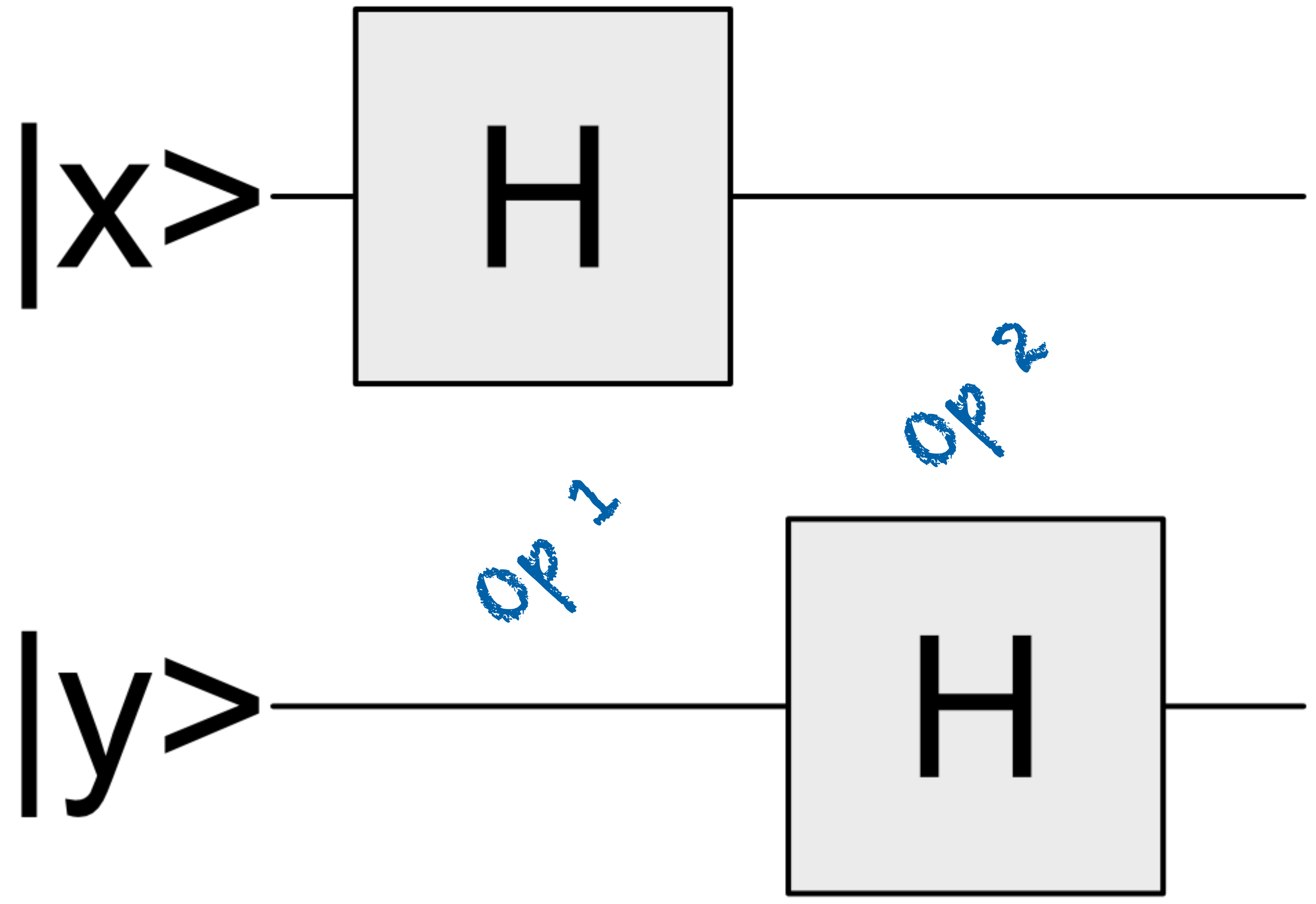


Entanglement!

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}\left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\right) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

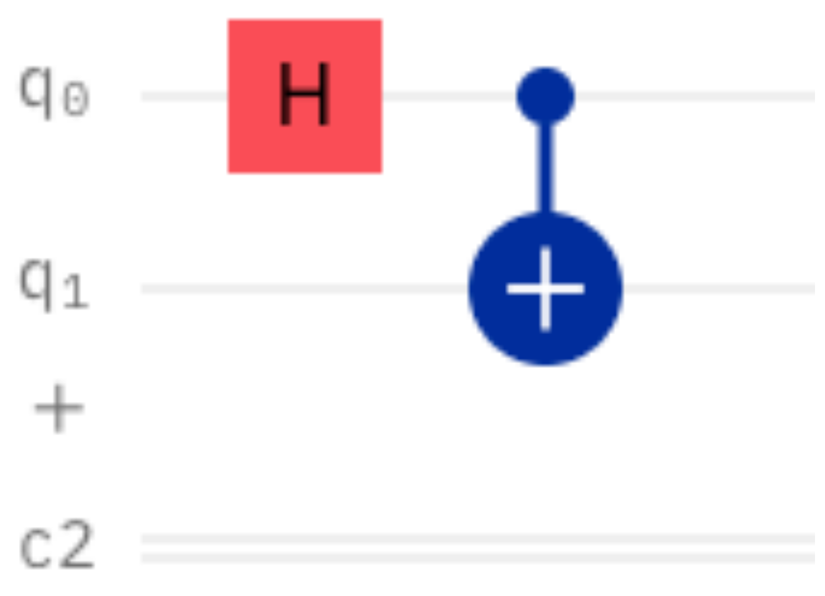
Circuit composer: https://quantum-computing.ibm.com/

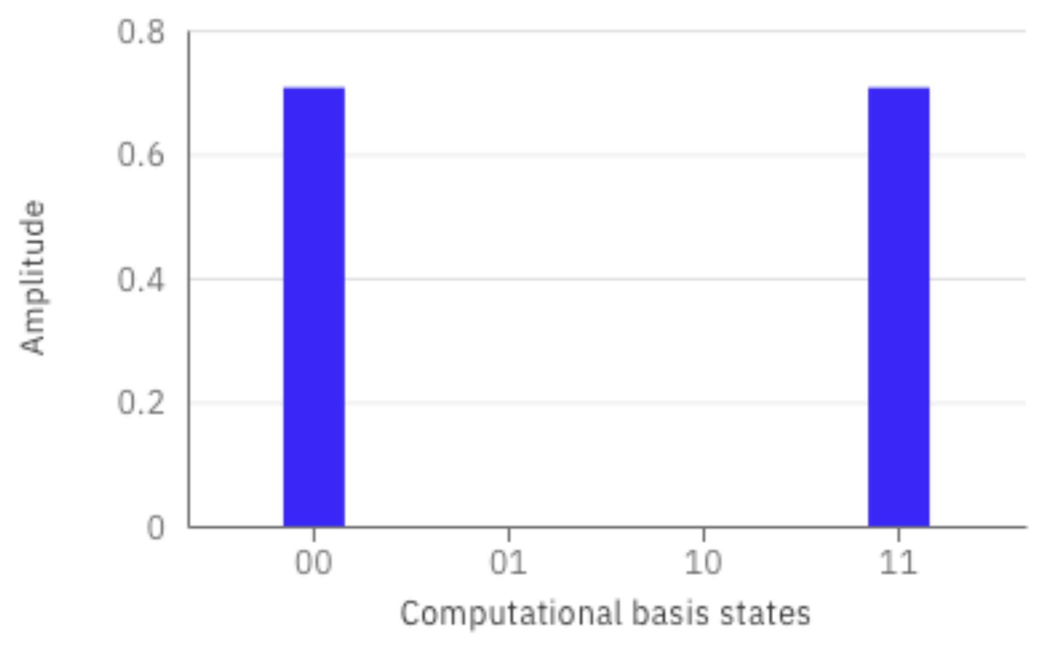
What *is* quantum computing?

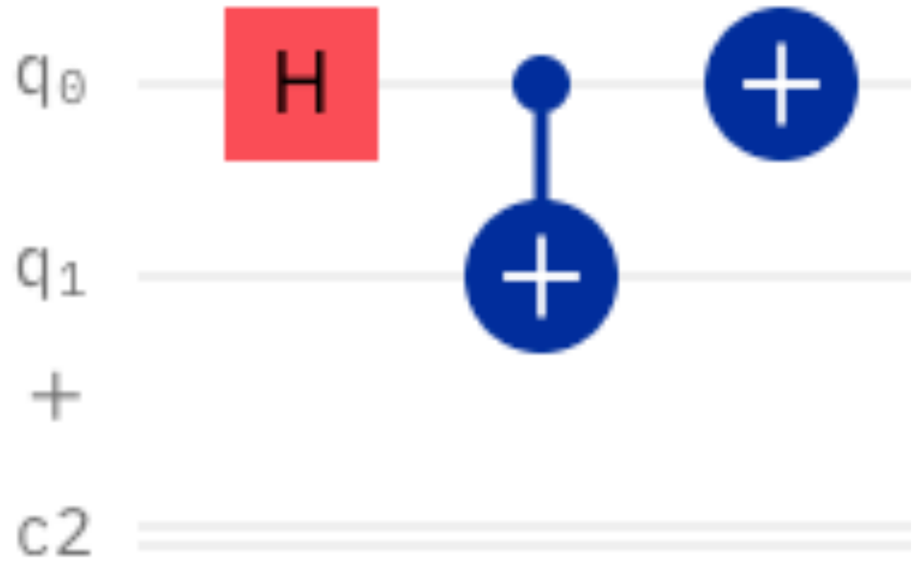


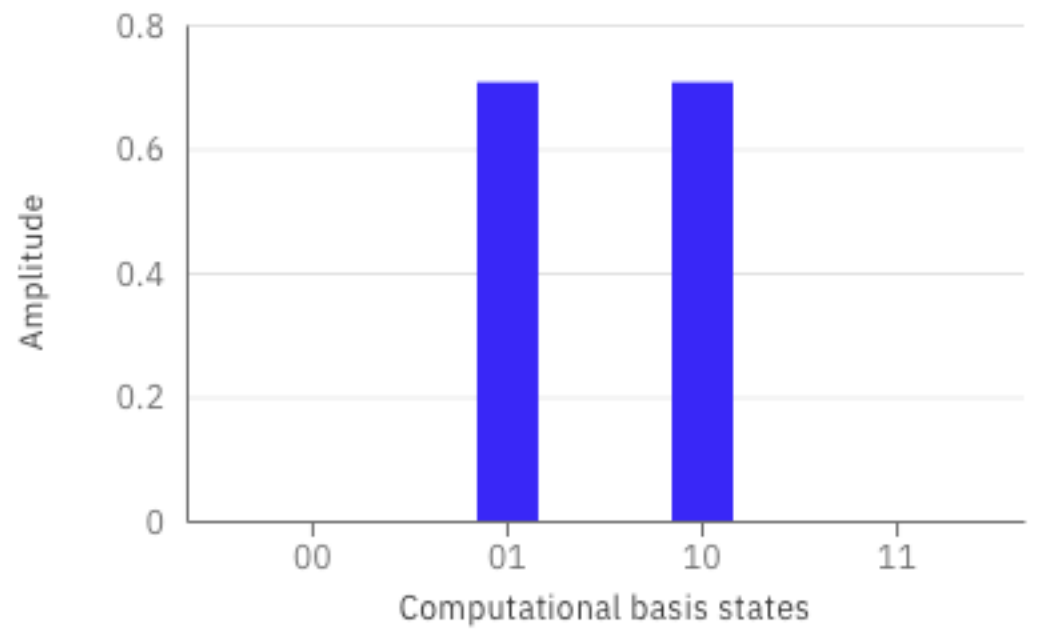
$$\underset{H \otimes I}{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}} \quad \underset{I \otimes H}{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}}$$

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$





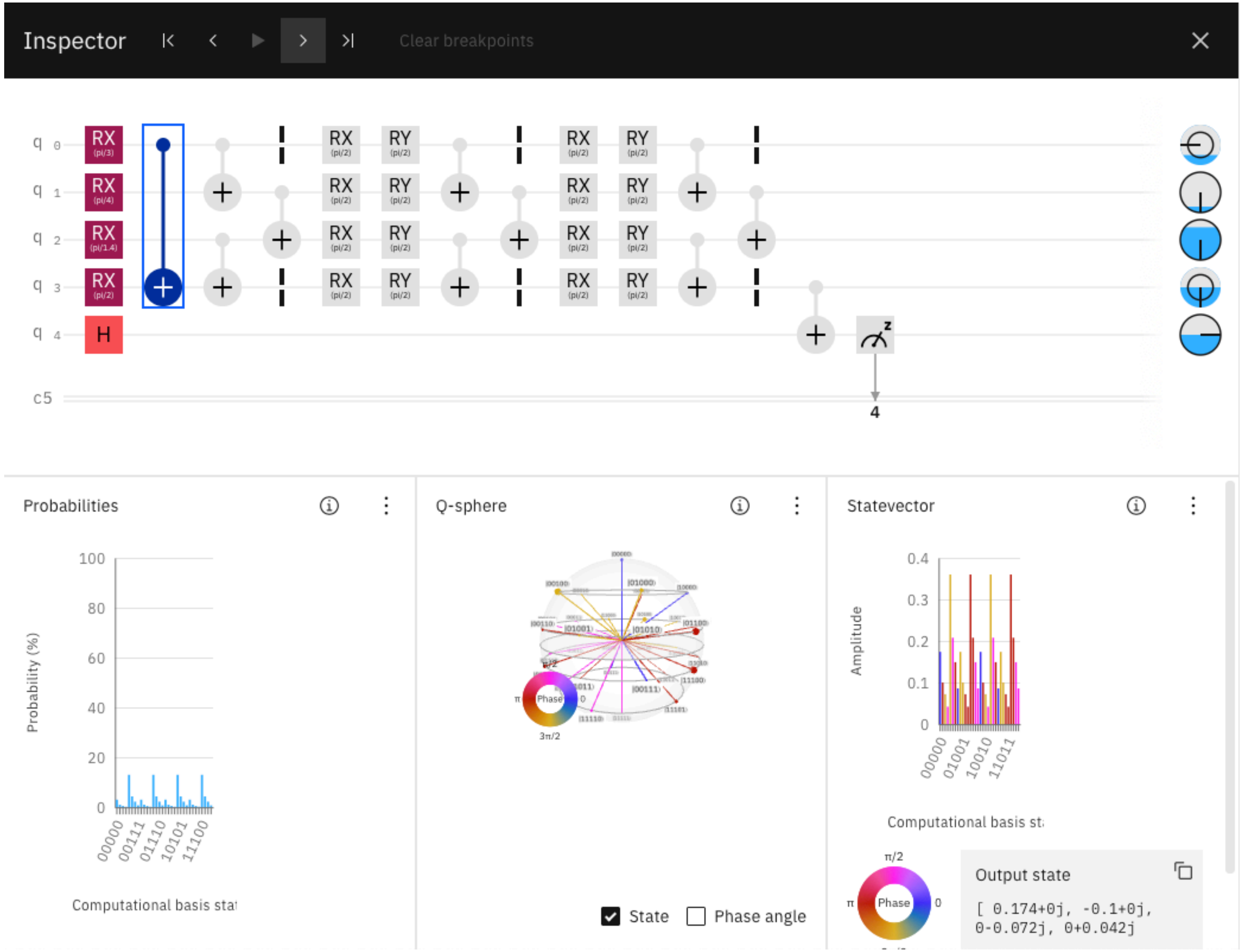




Super hand-wavy “quantum advantages”

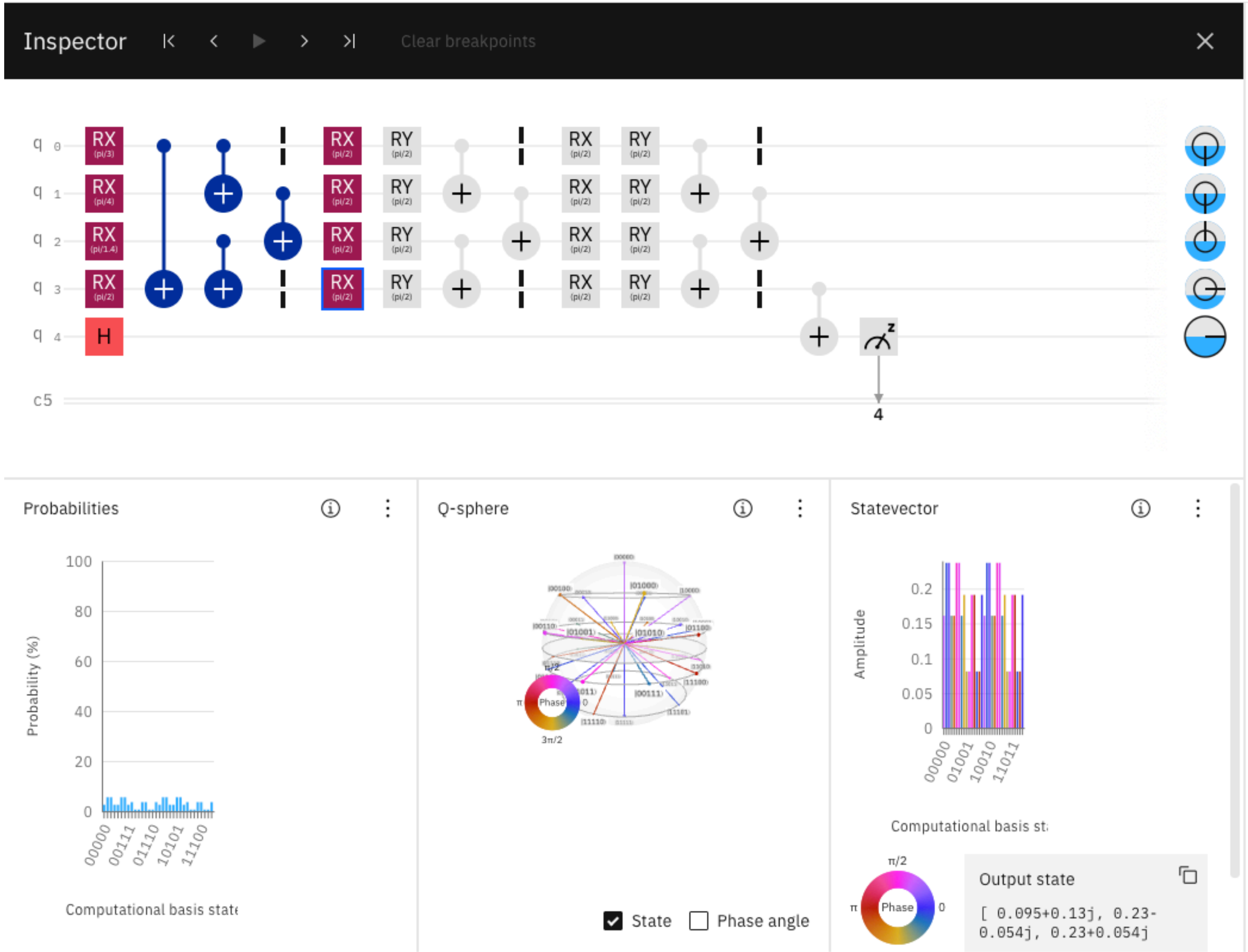
- Superposition lets us create a sum state with two operations instead of four.
- Entanglement means we can manipulate the entire state vector with one operation.
- *Exploiting* these operations with *provable* speedup is actually pretty hard! (Consider measurement if nothing else…)

The Quantum Circuit



https://quantum-computing.ibm.com/composer

The Quantum Circuit

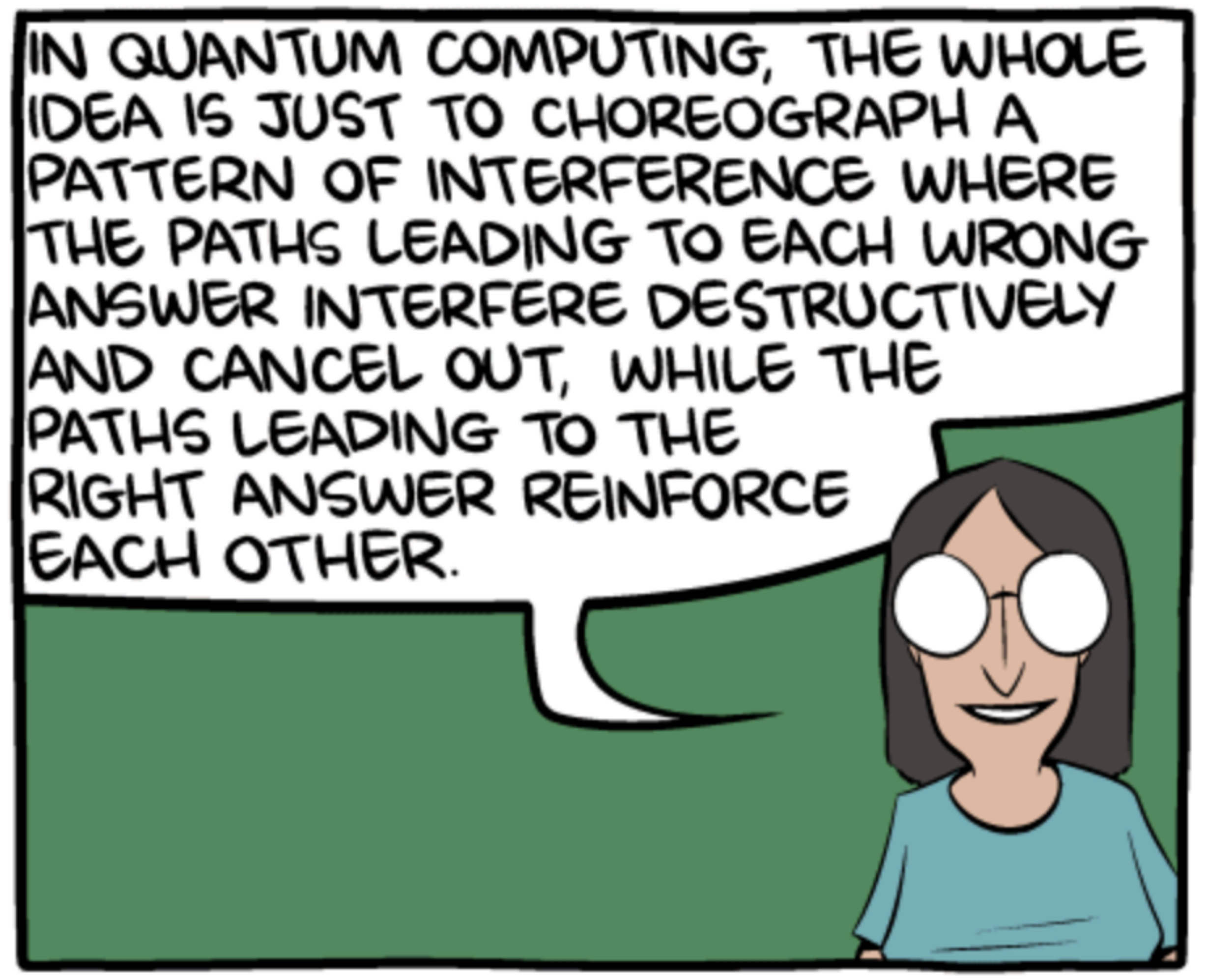


https://quantum-computing.ibm.com/composer

What is quantum computing *good for?*

Photo by Erik Lucero, Google

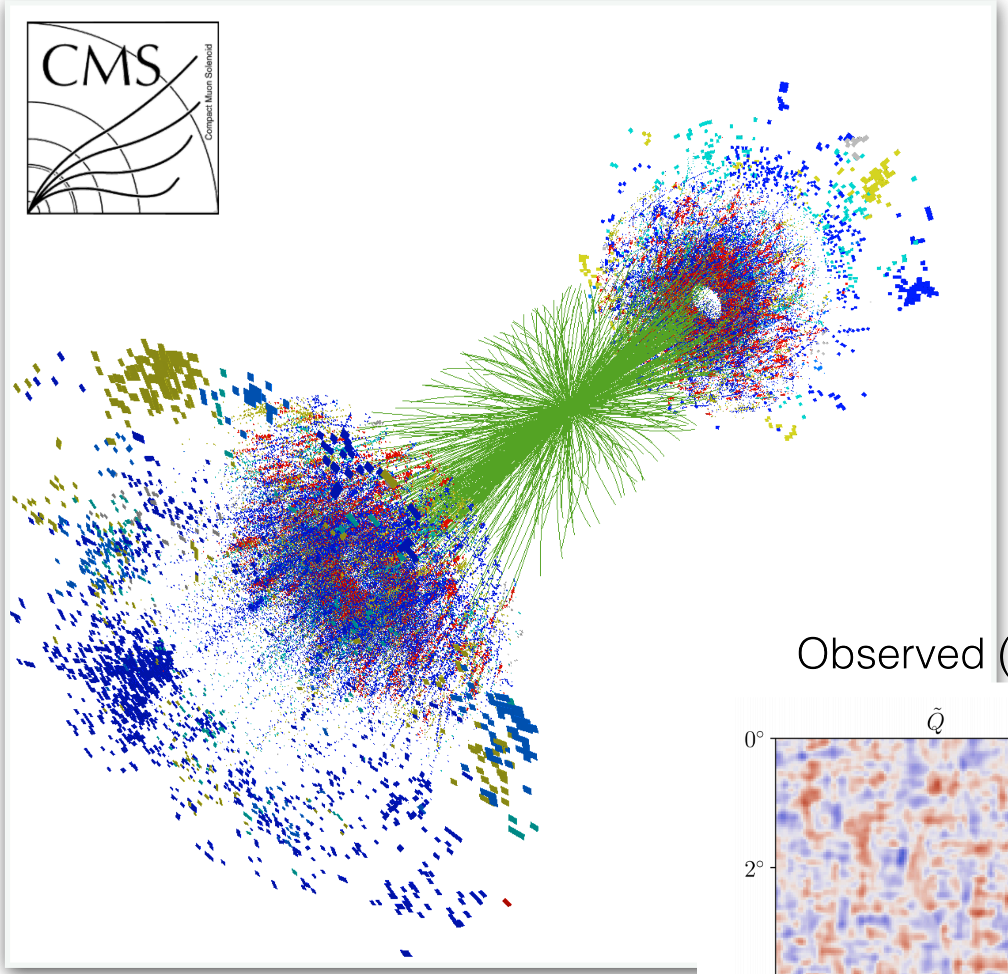
- Many things (cryptography, communications, etc.), but the "commercial killer app" will probably be the first proposal*: the simulation of quantum systems - and the money is in chemistry now. Quantum computers will ultimately be able to do something classical computers will never be able to do - simulate exactly the behavior of molecules with complex electron behavior.
- The physics undergirding this is that of a system of interacting fermions.
- There are fewer commercial applications in the simulation of, say, nuclear matter in neutrino-nucleus scattering, but we can benefit from the commercially motivated research in quantum chemistry a great deal!
- Why is quantum computing powerful?
  - https://www.smbc-comics.com/comic/the-talk-3

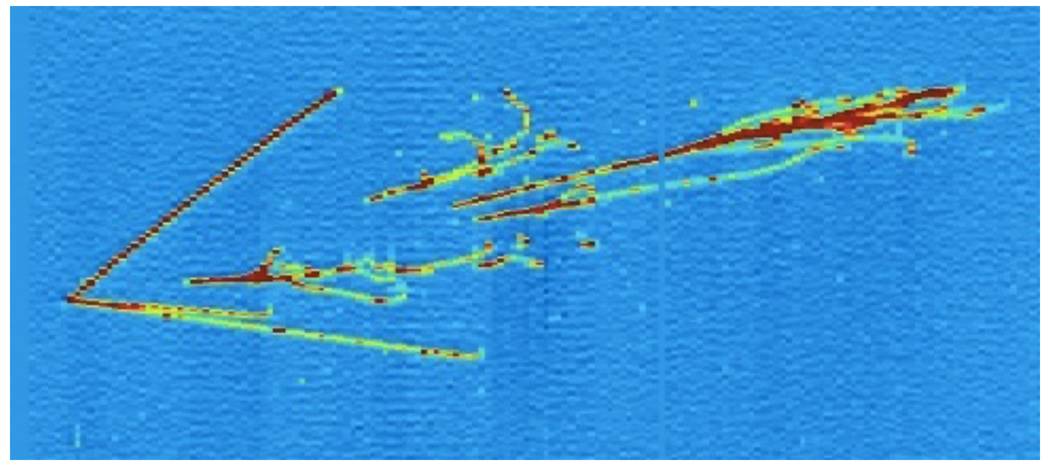


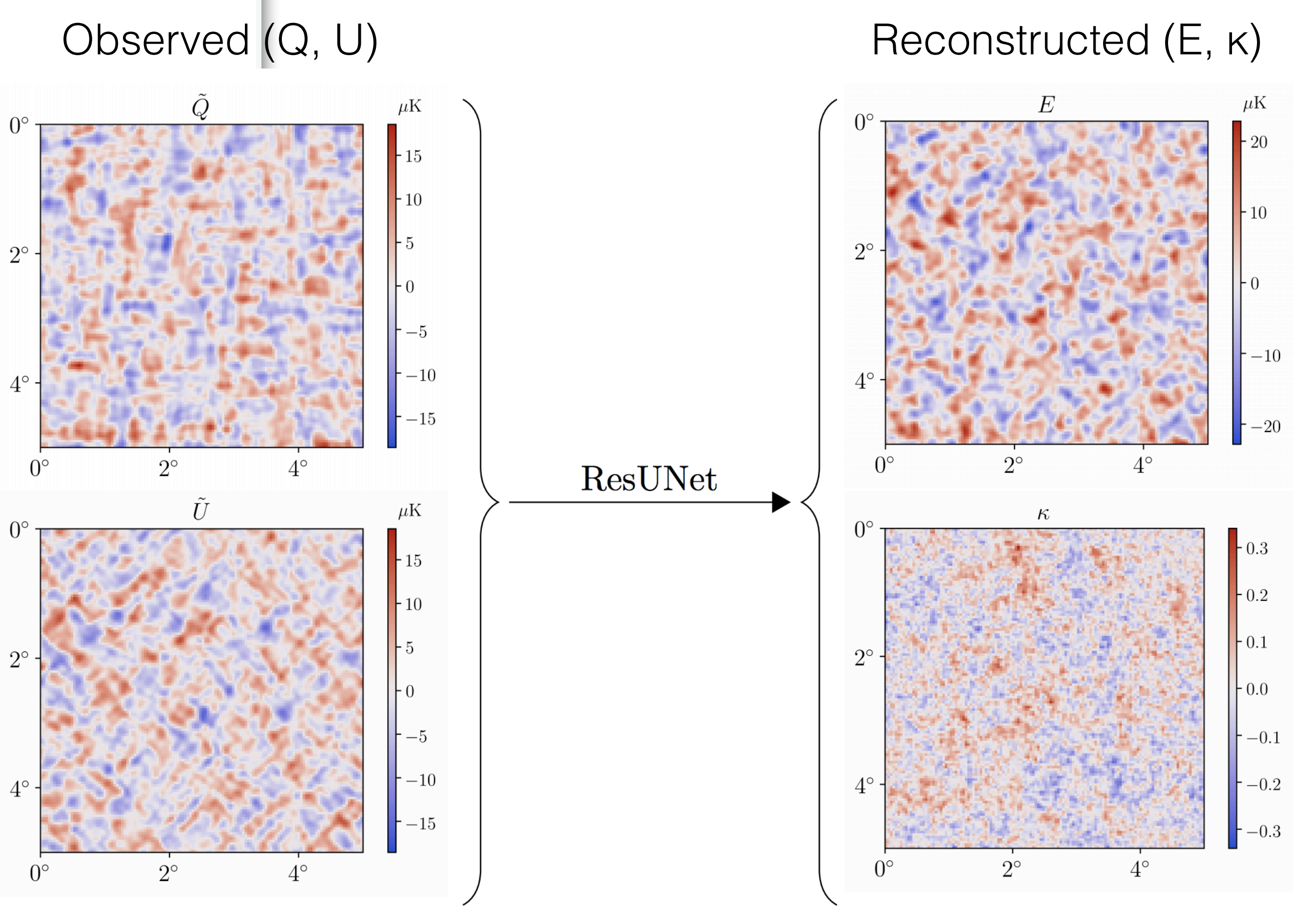
* R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, Jun 1982.

Machine learning for science

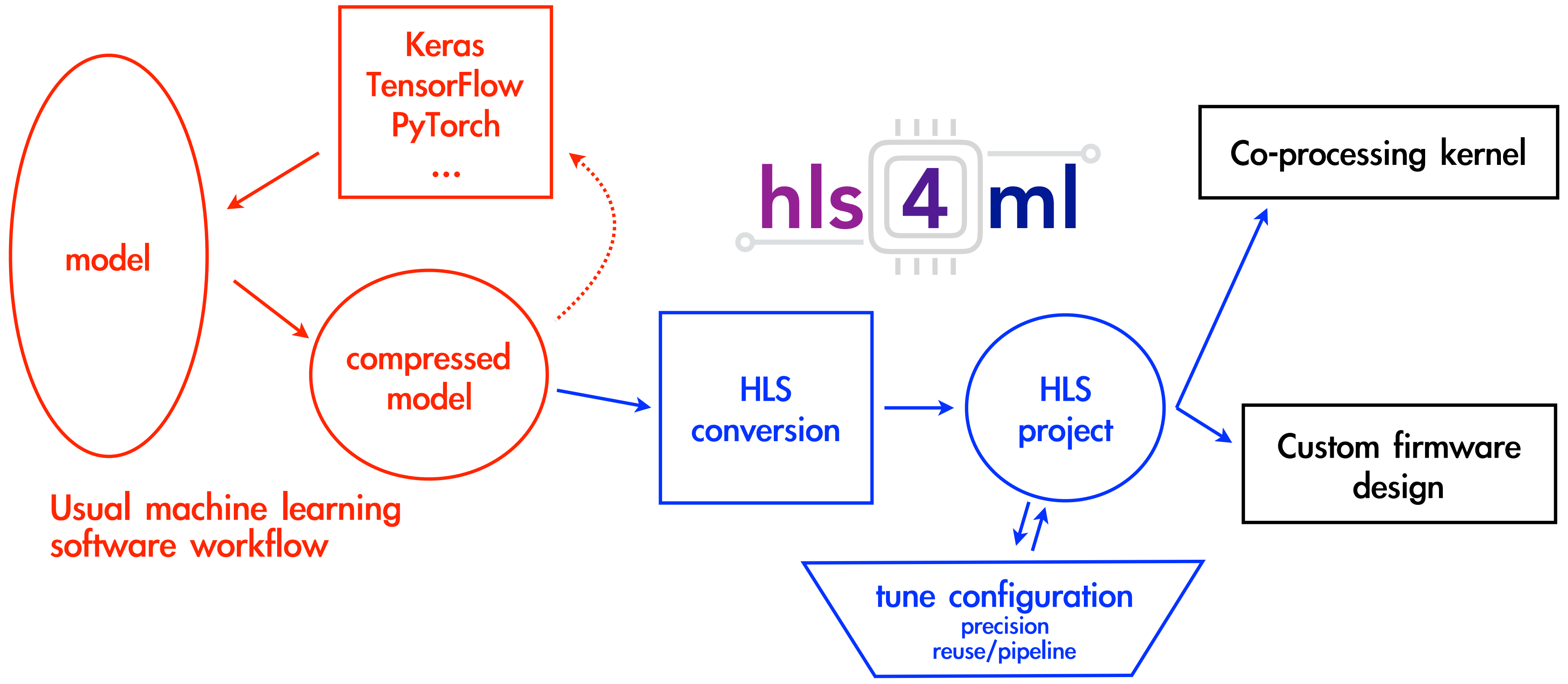
- Machine learning techniques have proven to be very powerful in scientific data analysis:
  - Neural nets where feature engineering is hard,
  - SVMs, gradient boosted trees, etc. where feature engineering is easy.
- Many applications in High Energy Physics (HEP) and Cosmology at Fermilab.
  - "Event reconstruction" (advanced feature engineering)
  - Signal / background separation
  - Physics parameter extraction
  - "Fast ML" for triggering
- ML shows up almost everywhere…







J. Caldeira, B. Nord, et al., https://arxiv.org/abs/1810.01483



https://fastmachinelearning.org/hls4ml/

Machine learning for science

- Machine learning techniques have proven to be very powerful in scientific data analysis:
  - Neural nets where feature engineering is hard,
  - SVMs, gradient boosted trees, etc. where feature engineering is easy.
- Many applications in High Energy Physics (HEP) and Cosmology at Fermilab.
  - “Event reconstruction” (advanced feature engineering)
  - Signal / background separation
  - Physics parameter extraction
  - “Fast ML” for triggering
- ML shows up almost everywhere…

https://fastmachinelearning.org/hls4ml/

Fermilab

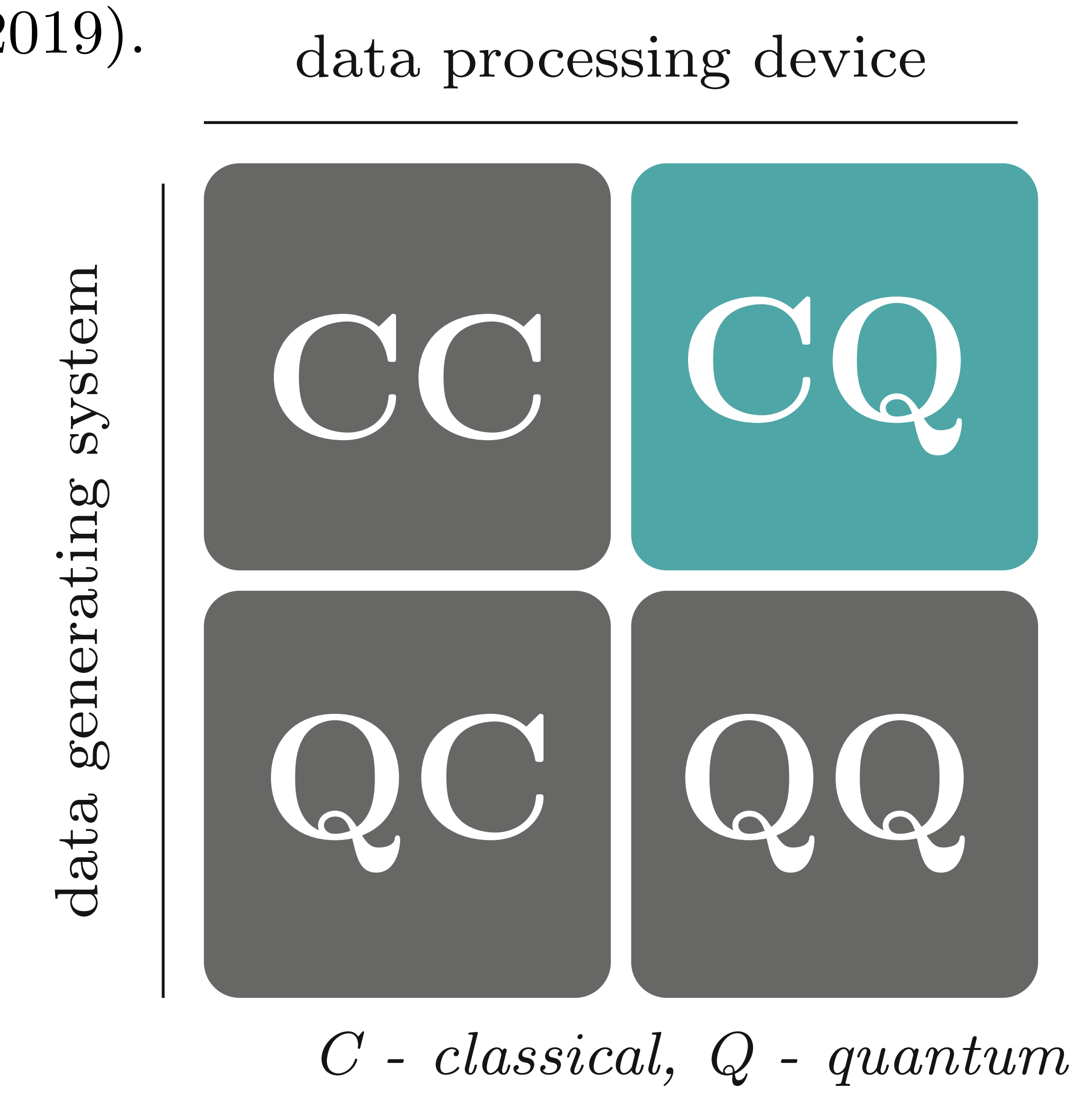
Quantum machine learning

- Science mission: to explore the application of “quantum machine learning” to *scientific data*.
  - Structured
  - Data generation process may be messy / unknown / the goal of a science effort.

- “Quantum ML” means different things depending on what the data source and algorithm’s physical substrate are.
- Most experimental and theoretical work in QML focuses on using a quantum processor to analyze classical data.
- Analyzing quantum data on a classical machine usually becomes a control problem, or a program optimization problem.
- Analyzing quantum data with a quantum processor makes the most sense in the context of analyzing the output of quantum sensors or the output of another quantum computer - we can’t store entangled states for long periods of time!

[1] V. Havlícek, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[2] M. Schuld and N. Killoran, Quantum machine learning in feature hilbert spaces, Phys. Rev. Lett. **122**, 040504 (2019).

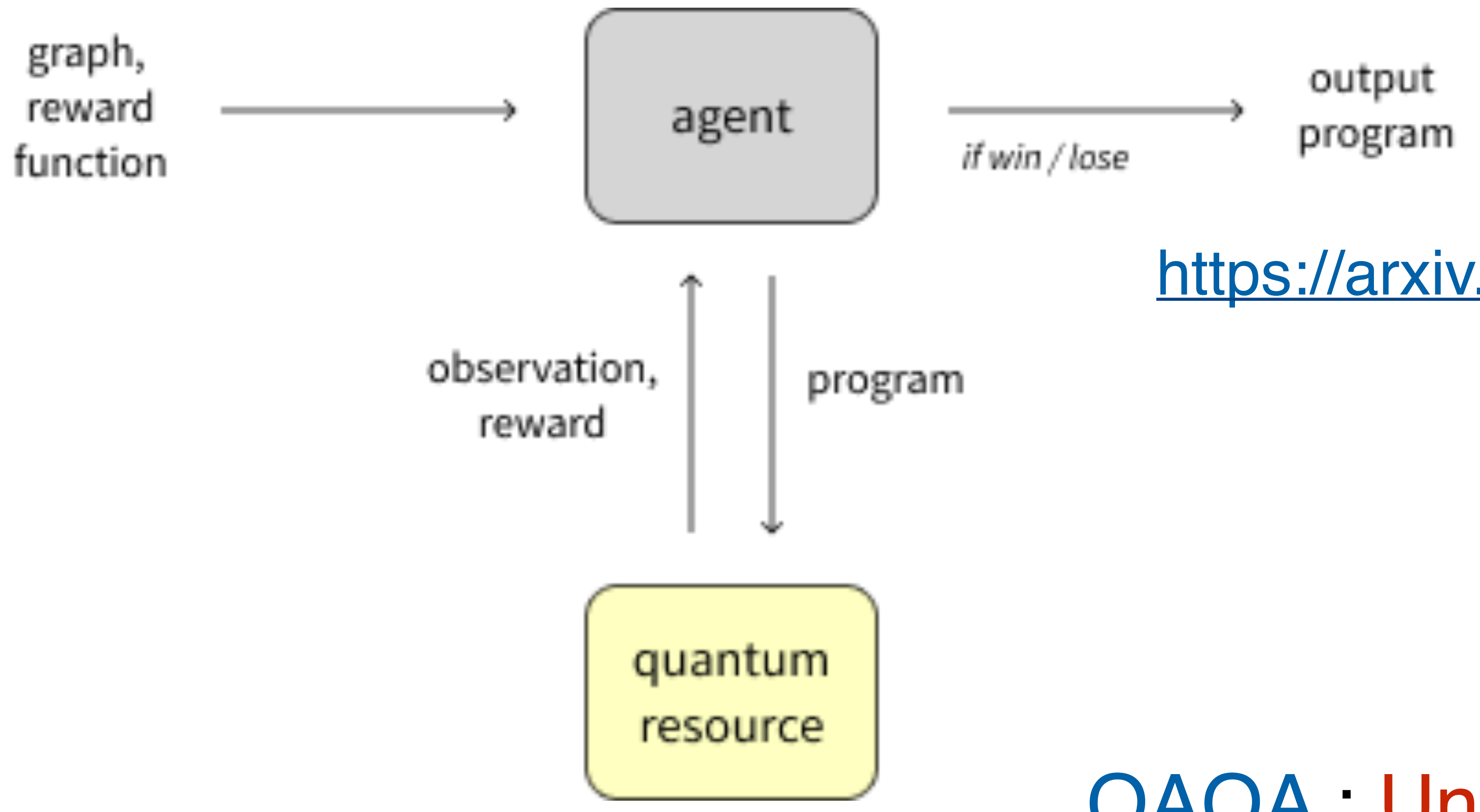


M. Schuld and F. Petruccione
Springer Press, 2018

Classical ML *for* quantum computing

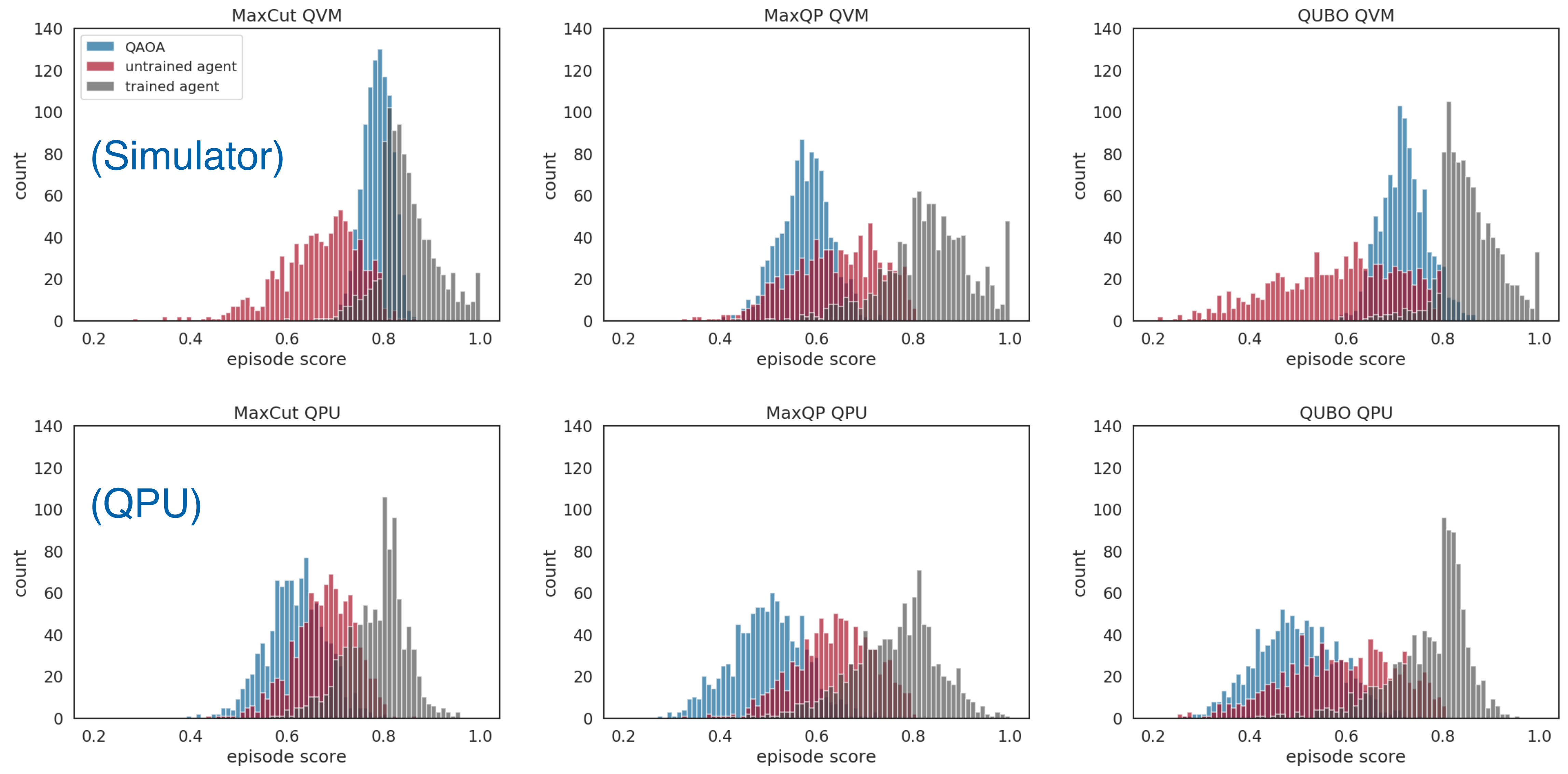
AUTOMATED QUANTUM PROGRAMMING
VIA REINFORCEMENT LEARNING
FOR COMBINATORIAL OPTIMIZATION

- Quantum computers require exponential resources to simulate even at high levels of abstraction. Traditional HEP approaches (build a simulation, tune it perfectly, and use that as the base for analysis) don't scale well in the quantum regime.
- We need heuristic algorithms that are capable of capturing the essential details, and without the need to write down a physically-motivated model.
- ML excels at this application.
  - *Example* - building and compiling optimal quantum programs (circuits) is an NP-hard graph problem.
  - Researchers have successfully trained reinforcement learning (RL) agents - the same essential algorithm used to train game-playing agents like AlphaZero - to build quantum circuits for combinatorial optimization problems.



https://arxiv.org/abs/1908.08054

QAOA : Untrained : Trained



Fermilab

Different eras of QML (emphasis on CQ QML) - Pre-NISQ

- "NISQ" - Noisy Intermediate Scale Quantum computers, a term offered by J. Preskill (https://arxiv.org/abs/1801.00862) - defines the "modern" era of quantum computers.
- In the pre-NISQ era (before we had hardware sizes larger than one qubit), all algorithmic work was necessarily theoretical. Researchers assumed all-to-all connectivity and the availability of an arbitrary number of error-corrected qubits when developing algorithms.
- One early approach leveraged the Harrow-Hassidim-Lloyd (2009) algorithm for a quantum solution to a linear system of equations (quantum - polylog(N), classical - linear in N, certain conditions required). Here the QPU acts to greatly speed up a subroutine in a fundamentally classical algorithm.
- Another early approach leveraged the quantum adiabatic algorithm (annealing, tunneling), eg. Farhi et al (2001), Nishimori (1998). In this approach we encode the problem into a Hamiltonian where the ground state holds the solution. We prepare the ground state via the adiabatic theorem, e.g. evolve:

$$H\left(s\right) = \left(1 - s\right) H_p + s H_0$$

Special thanks to Q. Zhuang (U. of Arizona) for slide inspiration.

Different eras of QML (emphasis on CQ QML) - Pre-NISQ

- Some issues with bringing pre-NISQ work to today:
- HHL
  - Requires error correction
  - Requires qRAM for data loading
  - Very interesting challenges from improved classical algorithms inspired by quantum breakthroughs (a common story!), e.g. E. Tang https://arxiv.org/abs/1807.04271, E. Tang https://arxiv.org/abs/1811.00414
- Annealing
  - When the problem is hard, the energy separation between states shrinks exponentially - not clear if quantum advantage is possible for interesting datasets and the problem gets much worse at finite temperature (e.g., D-Wave).
- See, e.g. 2017 review by Biamonte et al (Nature)

Fermilab

Special thanks to Q. Zhuang (U. of Arizona) for slide inspiration.

Different eras of QML (emphasis on CQ QML) - NISQ

- Black box - build it and see if it works. Example: variational quantum circuit

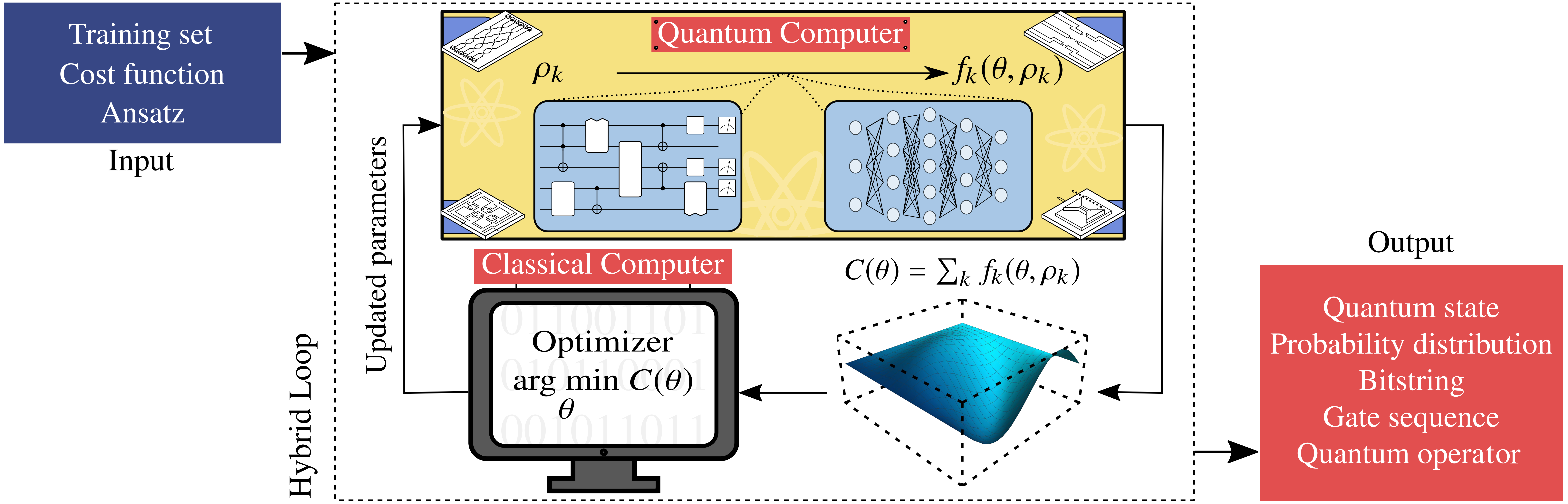


FIG. 1. **Schematic diagram of a Variational Quantum Algorithm (VQA)**. The inputs to a VQA are: a cost function $C(\boldsymbol{\theta})$ which encodes the solution to the problem, an ansatz whose parameters are trained to minimize the cost, and (possibly) a set of training data used during the optimization. At each iteration of the loop one employs a quantum computer to efficiently estimate the cost (or its gradients). This information is fed into a classical computer that leverages the power of optimizers to navigate the cost landscape and solve the optimization problem in Eq. (1). Once a termination condition is met, the VQA outputs an estimate of the solution to the problem. The form of the output depends on the precise task at hand. In the figure are indicated some of the most common types of output.

https://arxiv.org/abs/2012.09265

Special thanks to Q. Zhuang (U. of Arizona) for slide inspiration.

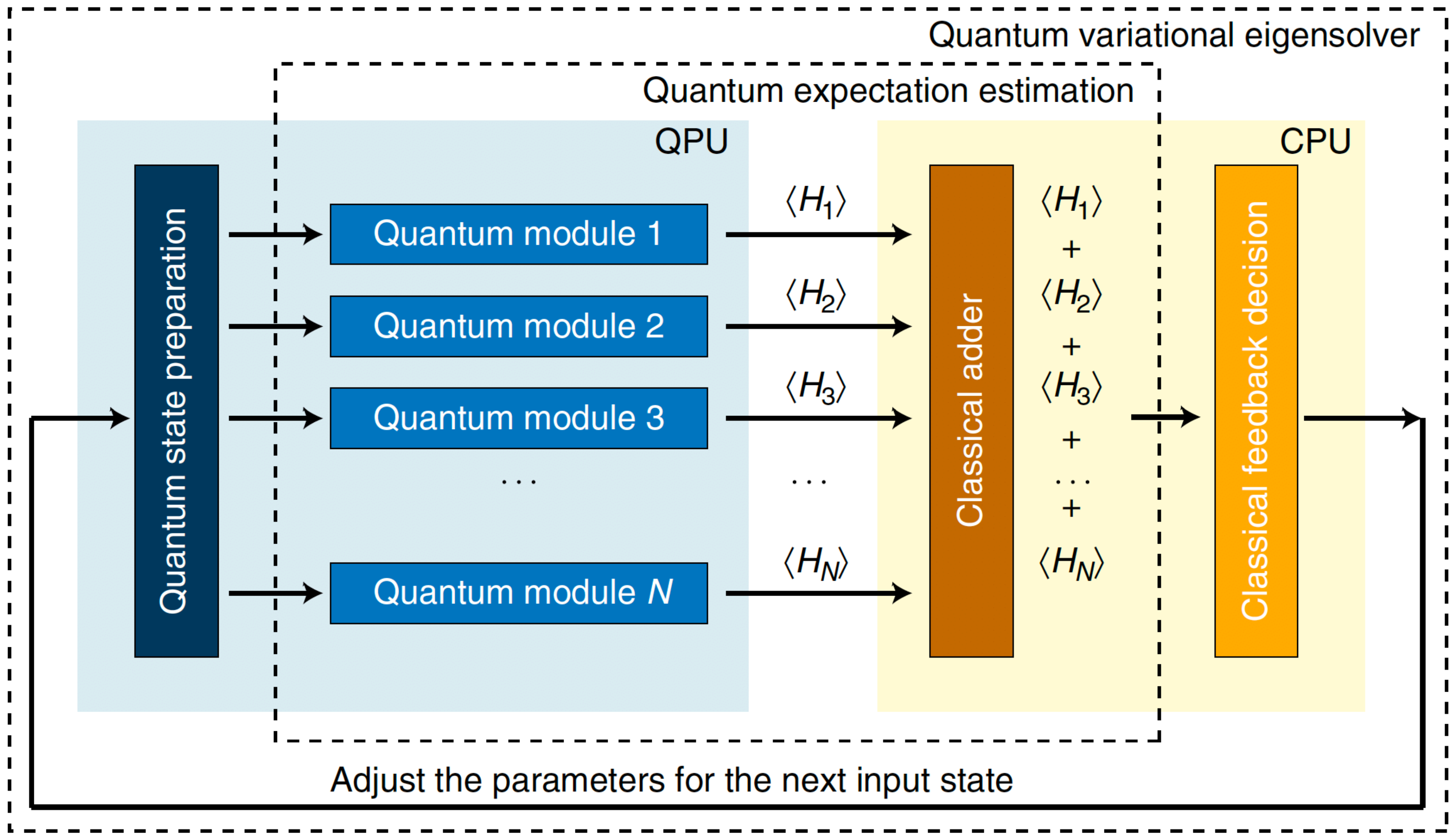
Different eras of QML (emphasis on CQ QML) - NISQ

- Black box - build it and see if it works. Example: variational quantum circuit

Eigensolver

e.g. Nat. Commun. **5**, 4213 (2014)
Nature **549**, 242–246 (2017)

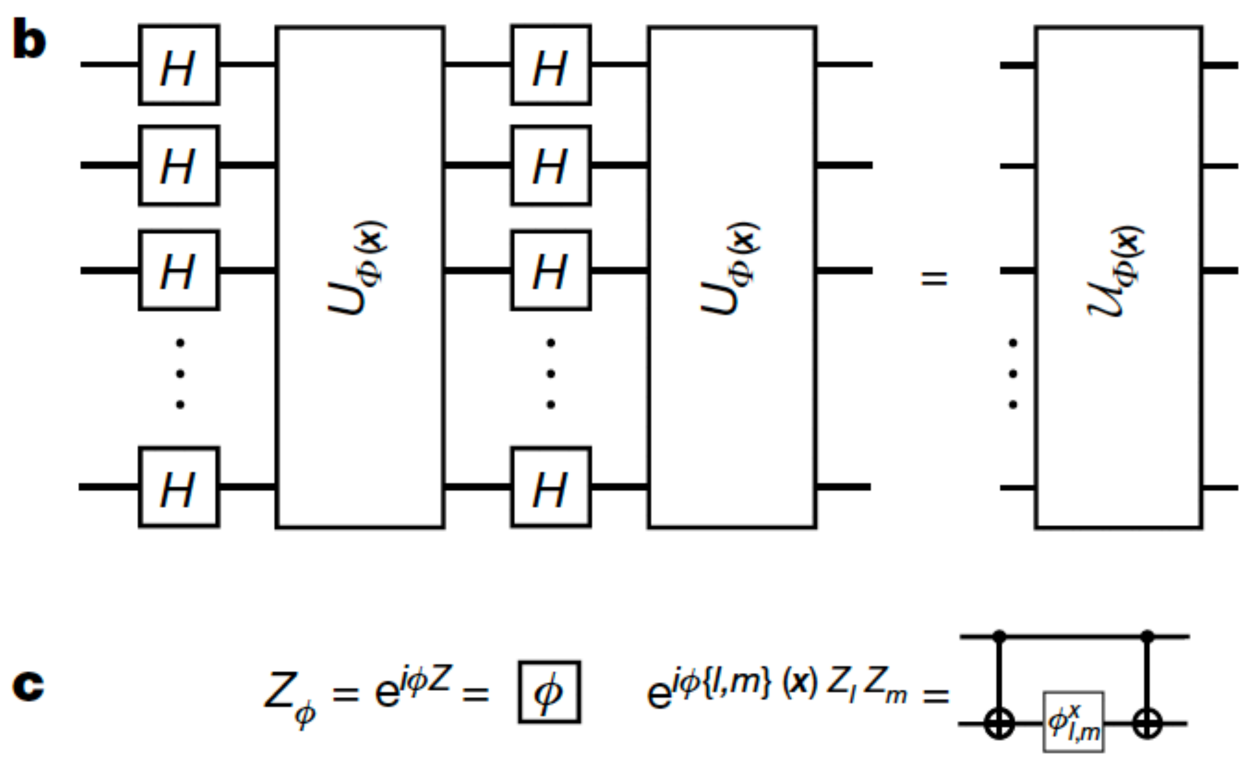
$$\mathcal{H} = \sum_{i\alpha} h^{i}_{\alpha}\sigma^{i}_{\alpha} + \sum_{ij\alpha\beta} h^{ij}_{\alpha\beta}\sigma^{i}_{\alpha}\sigma^{j}_{\beta} + \ \dots$$



Machine learning

e.g. Nature **567**, 209 (2019)

Encode data to VQC, inner product of the produced states provides kernel function between data---use quantum state as feature map: such functions can be hard to realize on a classical computer

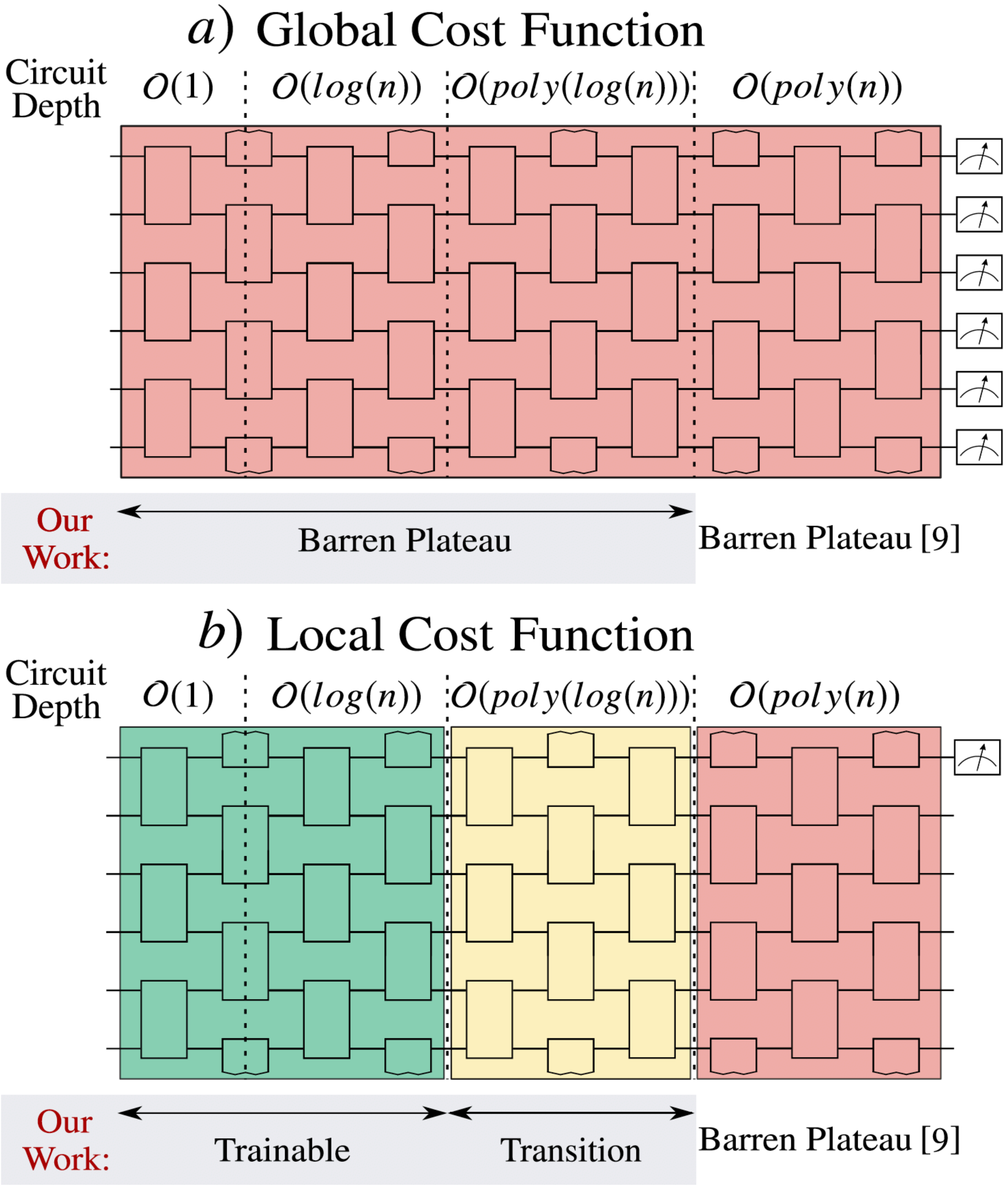


Special thanks to Q. Zhuang (U. of Arizona) for slide inspiration.

Fermilab

Problems with VQC

- Many, but an example - "barren plateaus"
  - Very easy to have a circuit ansatz where gradients are exponentially suppressed by the large Hilbert space. Optimization of the circuit parameters (already a very hard problem with no obvious path to scaling up) becomes impossible.
  - Roughly, log(depth) is the maximum we can sustain for a circuit to be trainable.
  - VQC likely cannot solve NP complete problems encoded on a Hamiltonian ground state.
  - For easy problems, the circuit may have structure and the barren plateau problem may be under control.
    - *See also: https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.2.010103 and*
    - *https://dx.doi.org/10.1038/s41467-021-22539-9*

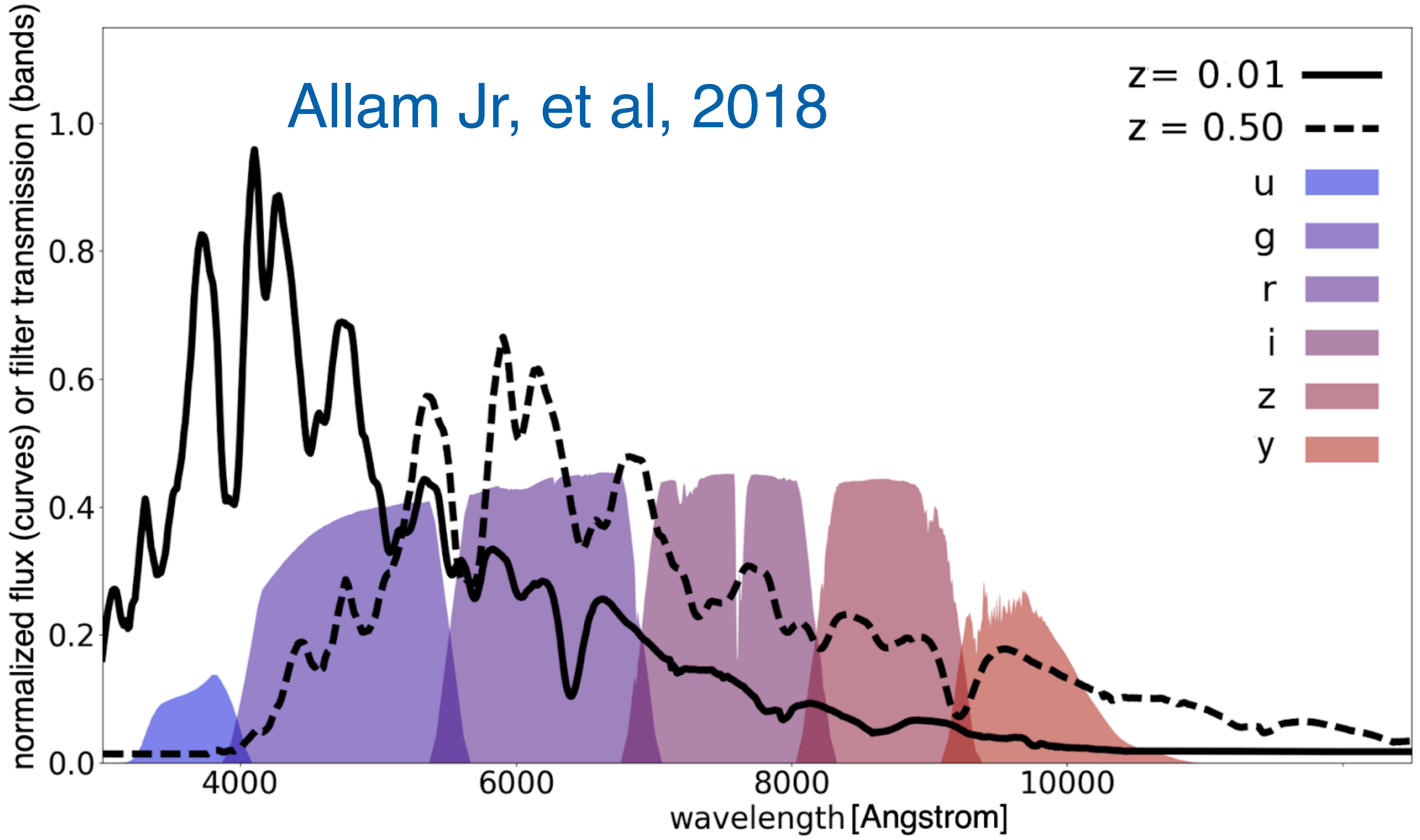


https://www.nature.com/articles/s41467-021-21728-w

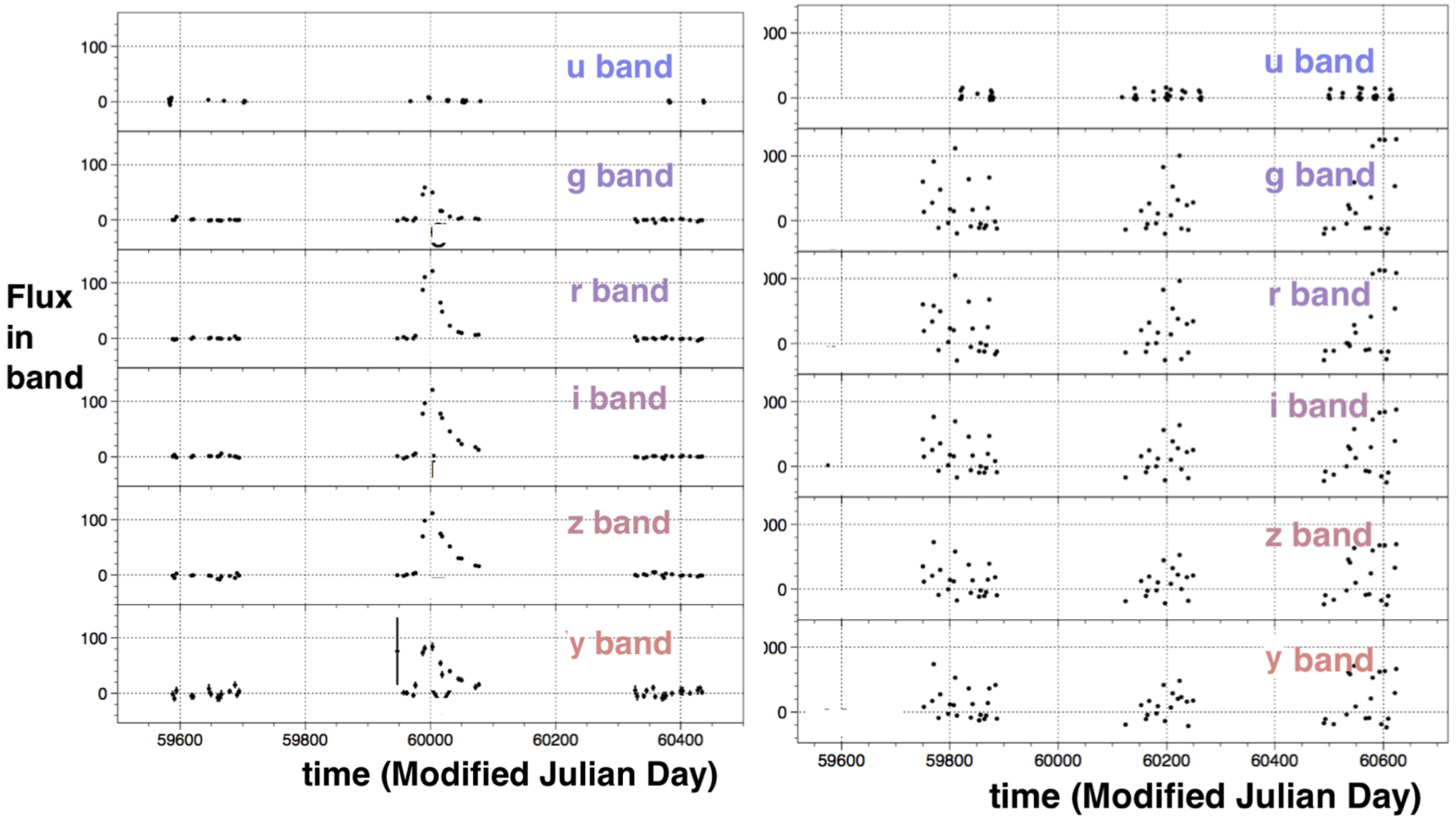
Special thanks to Q. Zhuang (U. of Arizona) for slide inspiration.

Example: Finding exploding stars with quantum computers

- Binary classification: Type II vs Type Ia supernovae (balanced 50/50 in this dataset).
- Dataset is time series values in different astronomical observational bands.
- These are Fourier transformed, and paired with distribution statistics (mean, skew, etc.) - 67 floating point numbers with some renormalization.
- **Same starting point as a science analysis - no preprocessing for dimensionality reduction.**



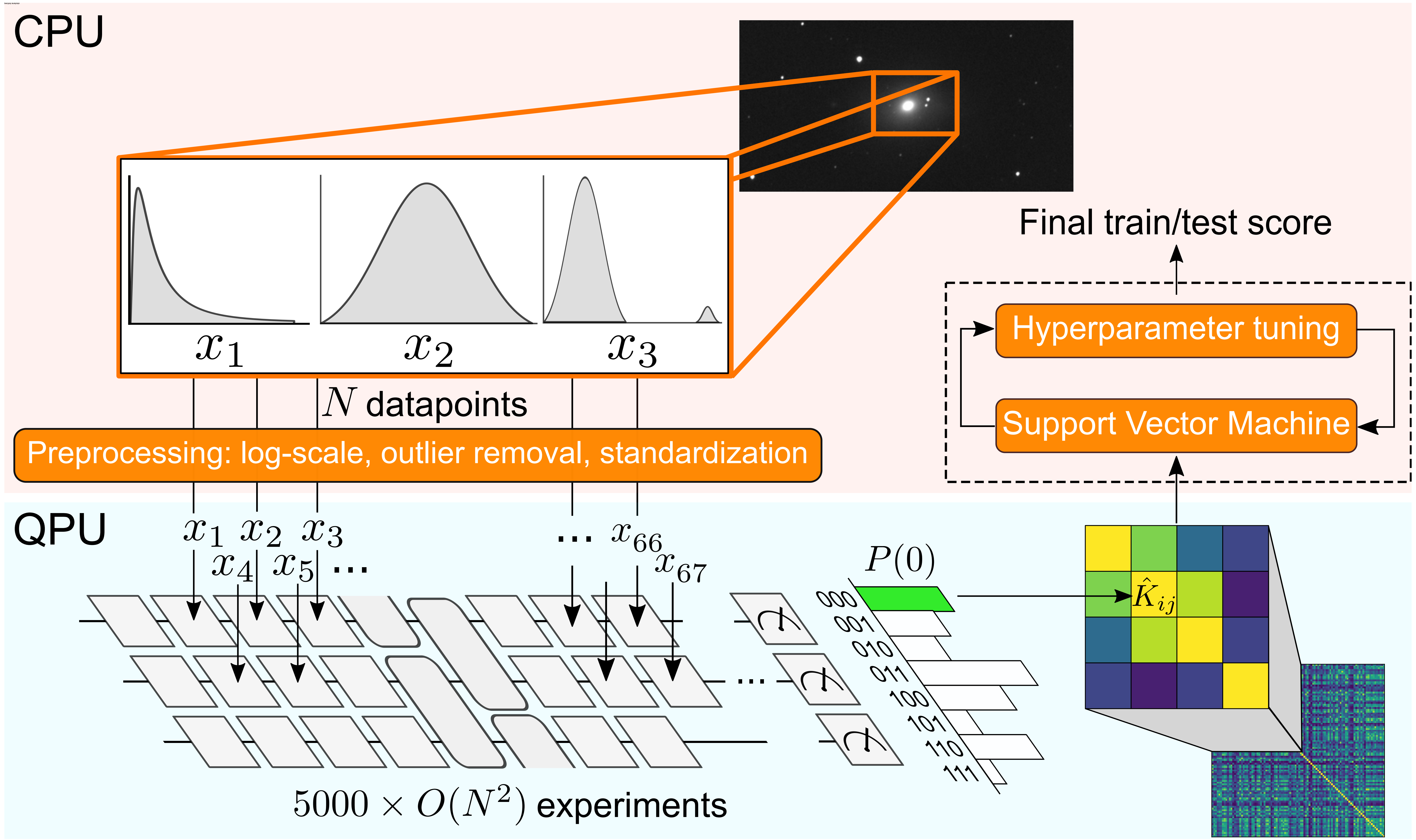
Two supernovae overlaid with ugrizy filter transmission.
Red-shifted event is further away.



Brighten and fade (left), repeated brightening (right).

Experiment workflow

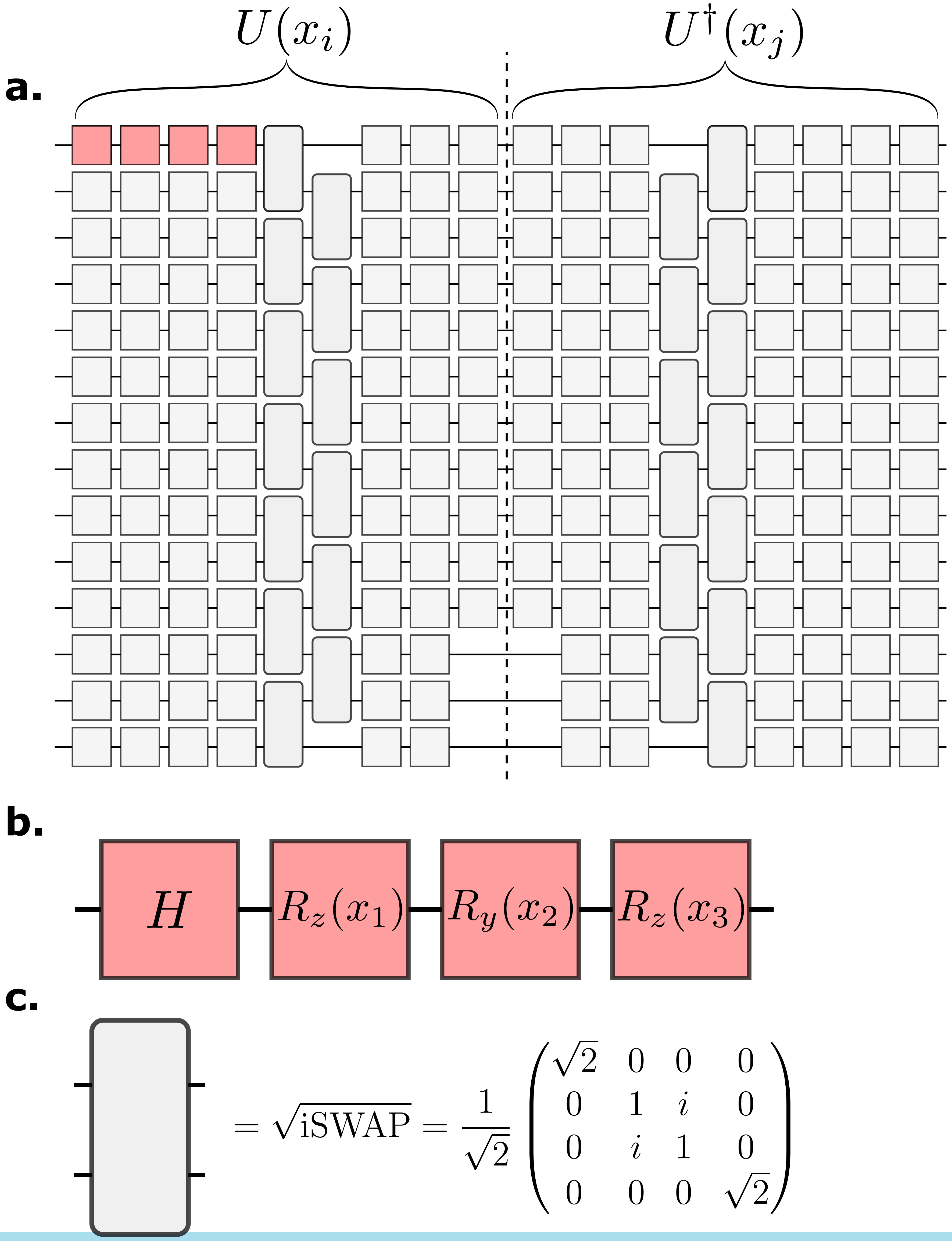
- Encode each event using qubit rotations, scramble with entangling gates.
- Compute UU† → effectively an overlap measure (probability of all 0's bitstring).
- Event vs event computation (so O(N²)) with dataset size).
- Feed the resulting matrix to a classical SVM.



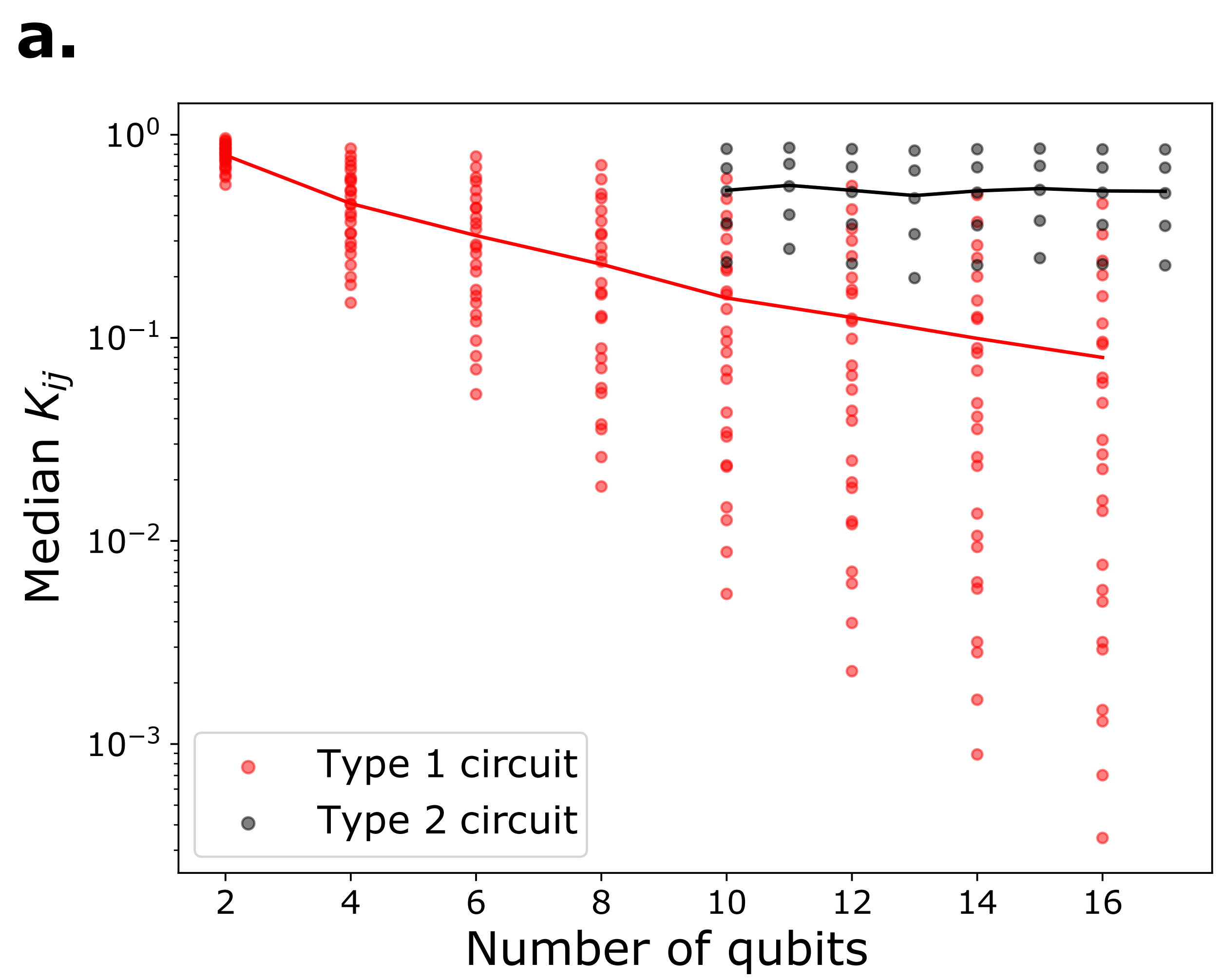
- See also: https://arxiv.org/abs/2105.02276
- https://github.com/thubregtsen/qhack
  - (And follow the references…)

Fermilab

Circuit Ansatz

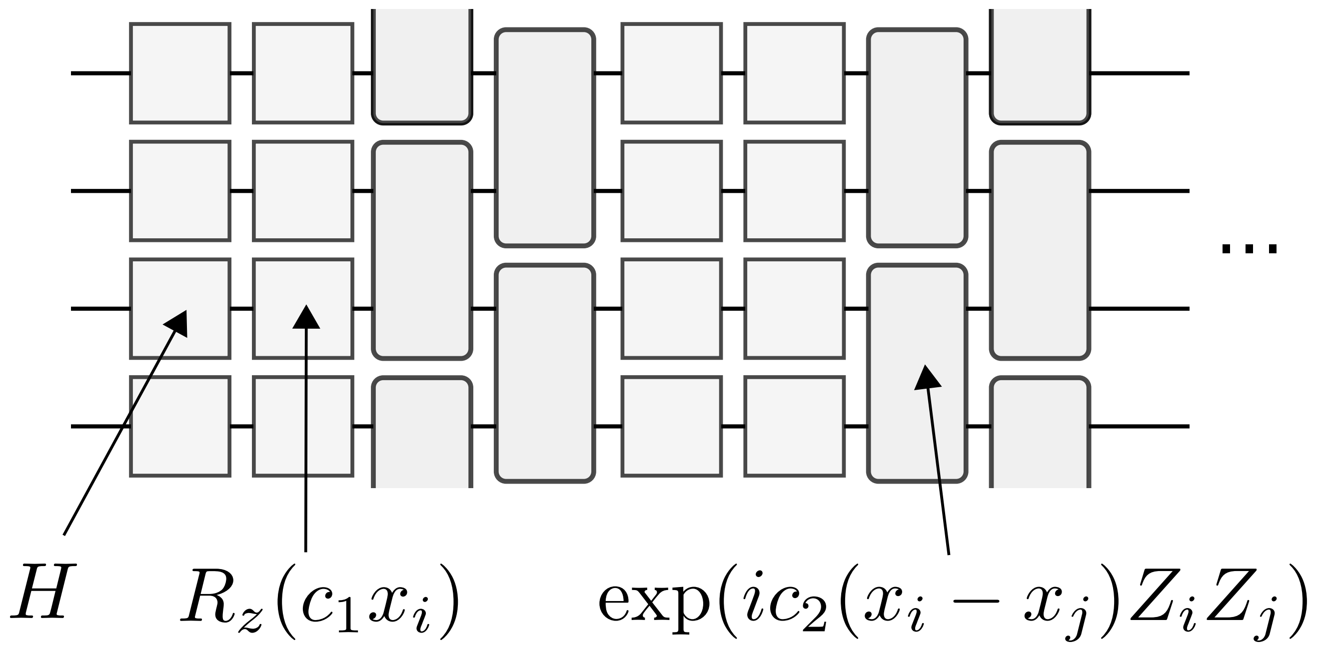


- Tested approaches in the literature at project inception - inner products (kernel elements) shrank with growing Hilbert space - $O(2^{-n})$.
  - Serious problem with a fixed shot budget.
- Avoid this by fixing the parameter count with respect to n.
- Statistical error still degrades SVM accuracy - empirically studied here.



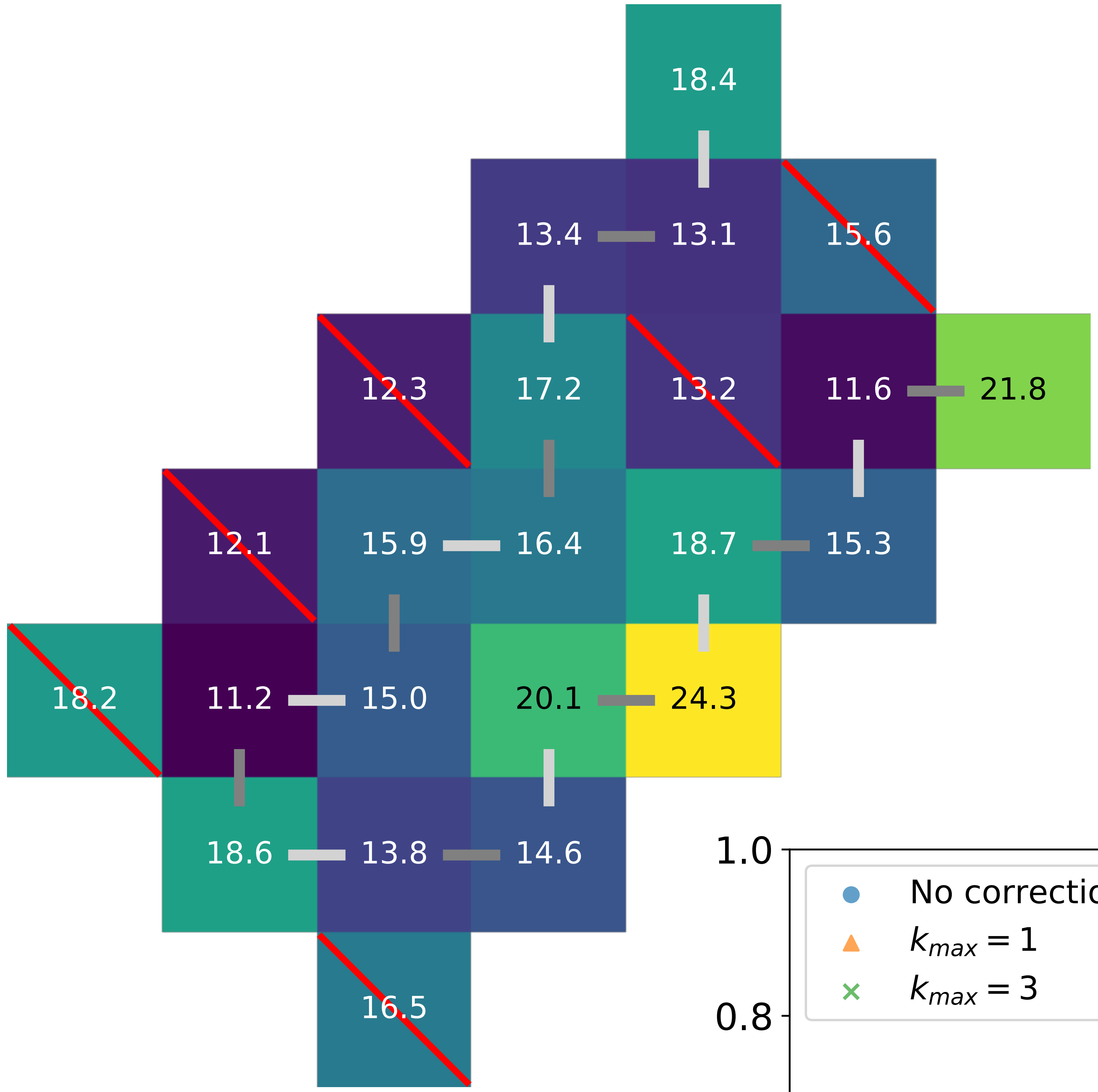
b.

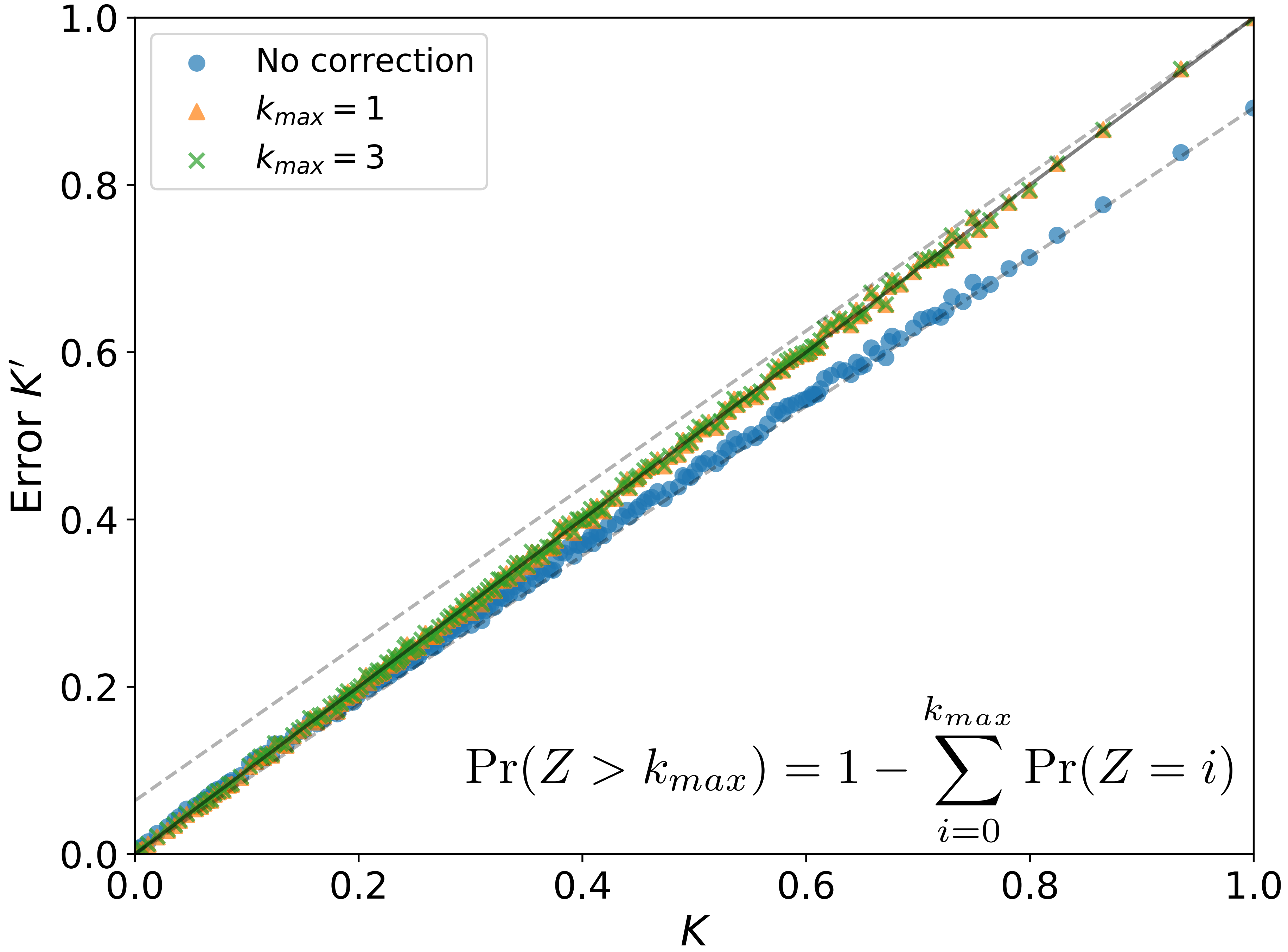
Note: Type 1 circuits required dimensionality reduction as data dimension was coupled to N_Q.



Error mitigation

- Some early projects (build a noise model and use simulation to “unfold”) did not scale for more than a handful of qubits.
- Qubit selection handled by an automated graph traversal algorithm with scoring function that weighted vertices and edges with calibration data (logarithmic function for T1, T2, plus cross-entropy benchmarks).
- Hamming-weigh truncated readout error correction (truncation required due to scaling problems in $O(2^n)$ matrix inversion).
  - Only needed to model at most one simultaneous readout error for a computationally efficient correction.
  - See https://arxiv.org/abs/2105.08161for an extension of this method.



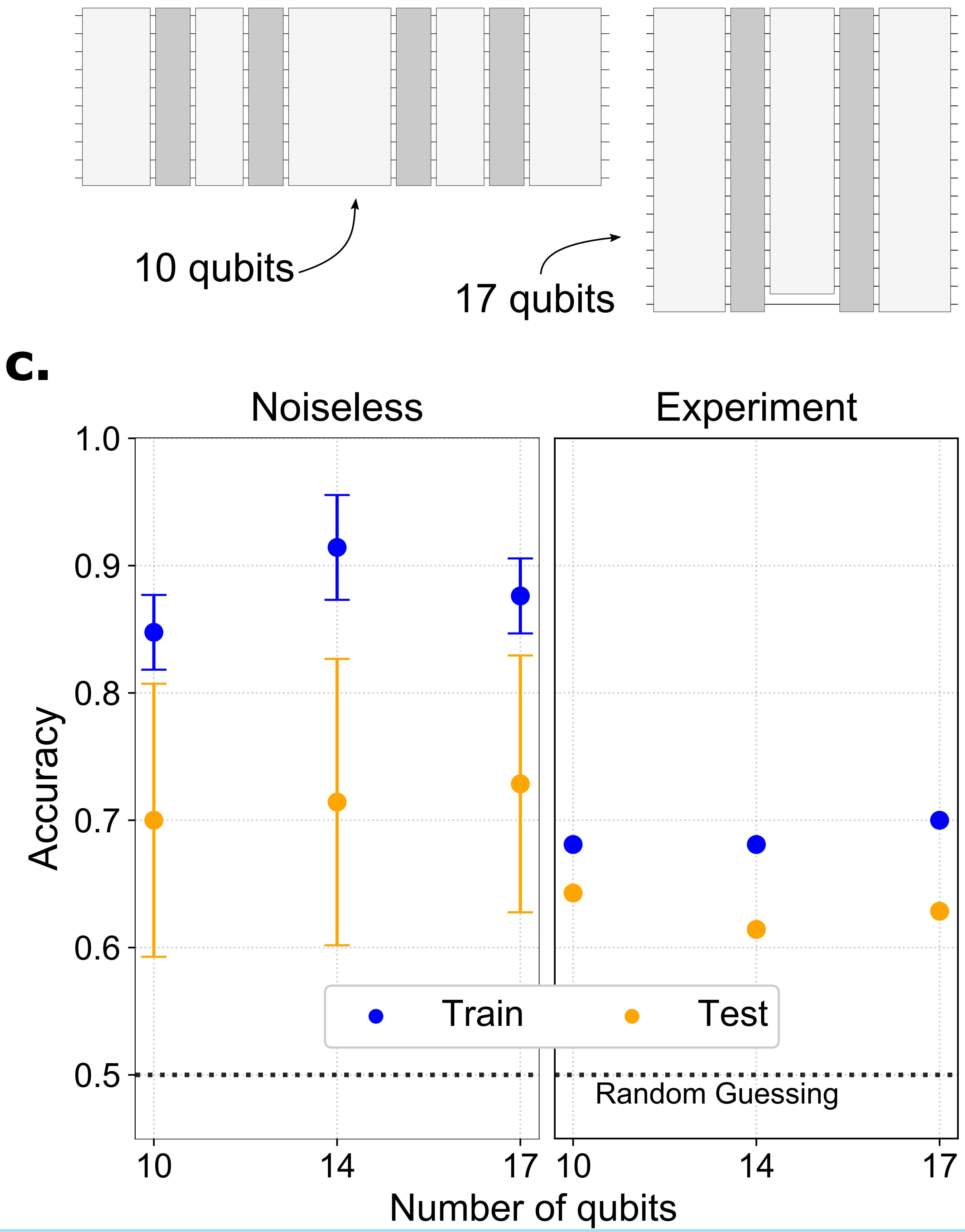


Fermilab

a.

Qubits	10	14	17
Depth	35(27)	23(17)	21(15)
$\sqrt{\text{iSWAP}}$ layers	8	4	4
$\sqrt{\text{iSWAP}}$ ct.	40	28	32
1q rot. ct.	254(174)	246(162)	270(170)

b.



Results

- Final classifier accuracy not driven by the number of qubits.
- Main advantage of higher qubit count (and increased depth) - encode data of higher dimension.
- Kernel classifier method shows intriguing intrinsic robustness against noise - even in cases where circuit fidelity was low we were able to achieve interesting classification accuracies.
- Competitive with noiseless simulation and classical benchmarks.

See also: https://arxiv.org/abs/2105.03406 for the new "qubit record" in a kernels algorithm plus some good ideas about what sort of dataset might be amenable to a quantum advantage.

Moving towards QQ QML…

Rethink about many machine learning tasks

One way to benefit from quantum

Data processing inequality: performance limited by measurement noise

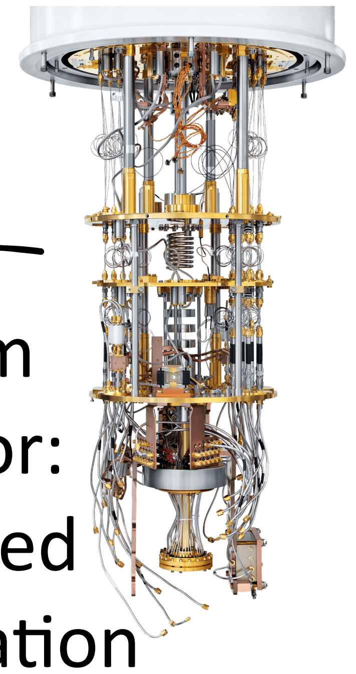
Light source

Reflection

Sensor

Classical data

Encode classical data to quantum: qRAM or VQC



Quantum Processor: supervised classification

Decision: it is a car

Rethink about many machine learning tasks

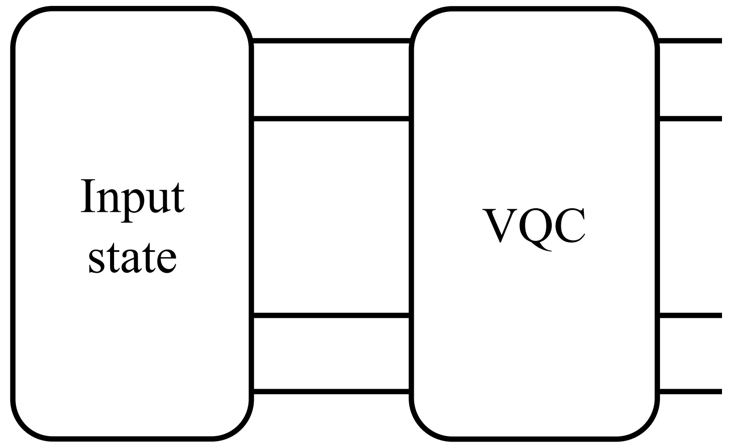
Our way of benefiting from quantum: the sensing process is described by quantum physics

Quantum light source

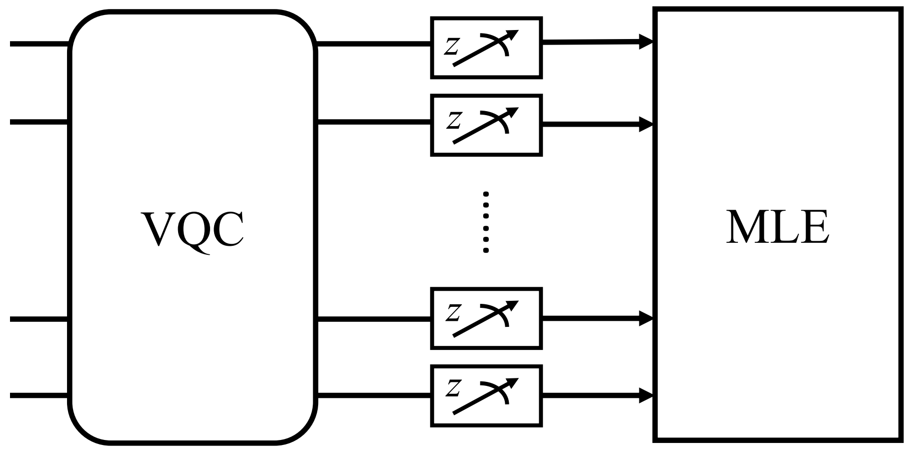
Reflection: modeled as quantum channels

Quantum sensors

Decision: it is a car



Quantum source prepared by a VQC



Measurement performed by a VQC

Figures courtesy of Quntao Zhuang (U. of Arizona)

Special thanks to Q. Zhuang (U. of Arizona) for slide inspiration.

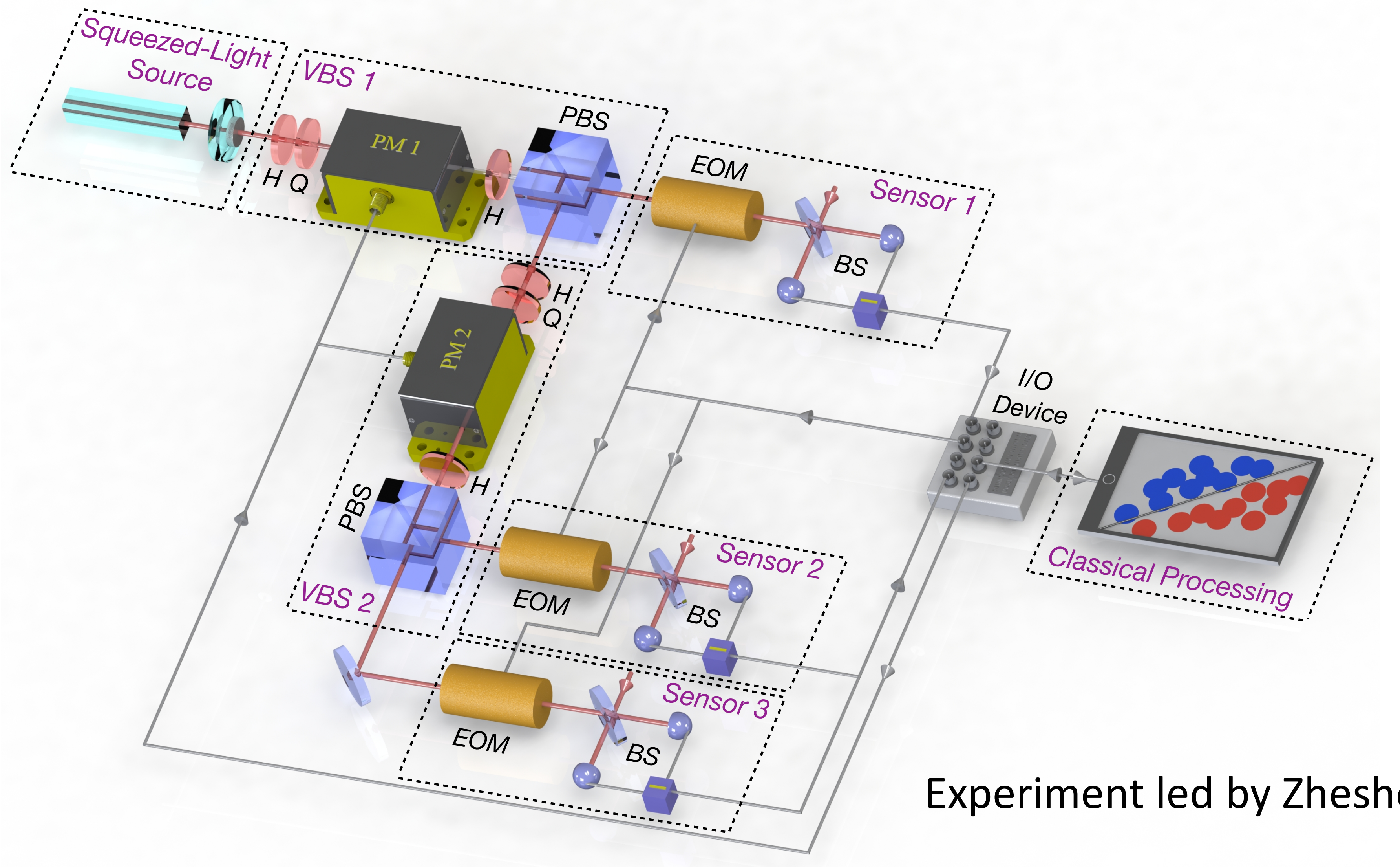
Supervised learning assisted by an entangled sensor network (SLAEN)

- See https://journals.aps.org/prx/abstract/10.1103/PhysRevX.9.041023 for theory.

Experiment demonstration

Phys. Rev. X **11**, 021047(2021)
Featured in Physics 14, 79 (2021)

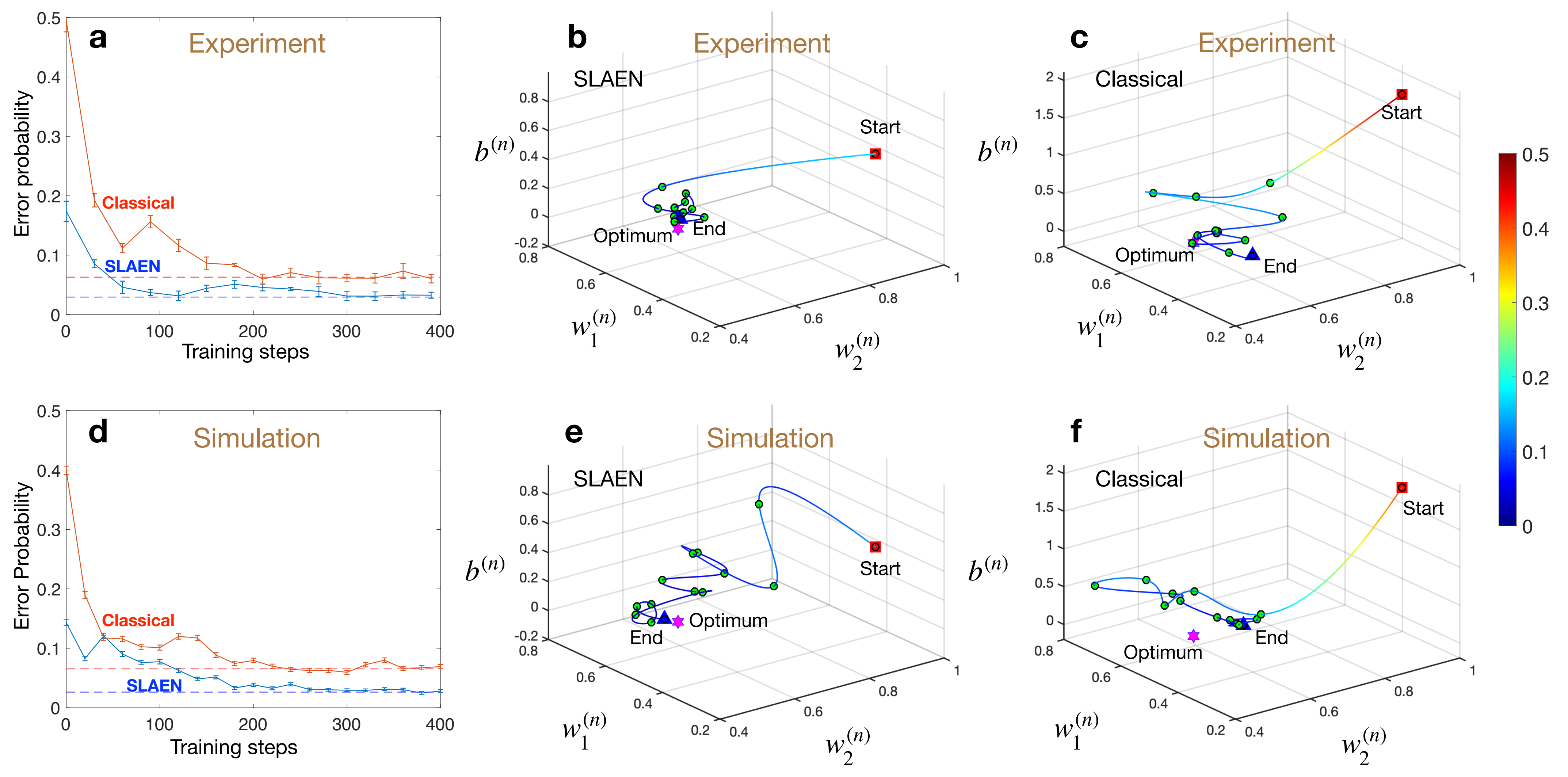
- Three sensors.
- Entanglement reconfigurable by beamsplitters.
- Measurement noise suppressed by the entangled distributed squeezed light



Experiment led by Zheshen Zhang

Experiment demonstration

Phys. Rev. X **11**, 021047(2021)
Featured in Physics 14, 79 (2021)



Special thanks to Q. Zhuang (U. of Arizona) for slide content and figures.

Conclusions

- Quantum computing is a powerful alternative to von Neumann architectures.
- It is specialized though - it has algorithm specific advantages and is limited by severe engineering challenges.
- Machine learning may prove to be a very interesting application space for quantum computing.
  - Quantum algorithms run on classical data are not very promising in the near term, although everyone is trying to do this (and this is the bulk of the literature, discussion).
  - Classical machine learning to help operate a quantum computer however, is extremely promising!
  - Even more interesting is quantum ML algorithms run on quantum data (output from another computer, or a network of sensors).
- Young minds must bring creative ideas, especially for utilizing quantum sensors!
  - Also promising: quantum generative networks.

Thanks for listening!