Quantum Computing for Machine Learners: A New Frontier

K.C. Kong
University of Kansas

Workshop for AI-Powered Materials Discovery at Great Plains
Jun 22 – 25, 2025

UNIVERSITY OF SOUTH DAKOTA

Image: Science News

Plan

- Brief History of quantum computing
- Machine Learning
- Why quantum computing
- $ vs tiger
- Quantum optimization
- An example from particle physics at Large Hadron collider

Presentations on quantum computing at this workshop

- From Bits to Qubits: A Beginner’s Journey into Quantum Computing (Tuesday, KC Kong)
- Quantum Computing for Machine Learners: A New Frontier (Tuesday, KC Kong)
- Enhancing quantum utility: Simulating large-scale quantum spin chains on superconducting quantum computers (Tuesday, Talal Chowdhury)
- Quantum Machine Learning Applications in High Energy Physics and Beyond (Wednesday, Konstantin Matchev)
- More talks on quantum physics
- Many talks on AI education and outreach program

What is a quantum computer?

- A quantum computer is a new kind of computer that's based on the laws of quantum physics.
- It can do certain things faster than normal computers because it follows a different set of rules.

Very brief history of quantum computing

"Zur Quantenmechanik" by Born and Jordan 1925

- 1925 The term "quantum mechanics" used by M. Born (Pauli, Heisenberg from U of Göttingen)
- 1925 Formulation of matrix mechanics by Heisenberg, Born, Jordan
- 1925-1927: Copenhagen interpretation
- 1930 "The principles of quantum mechanics" by Dirac
- 1935 Einstein, Podolsky and Rosen
- 1935 "Quantum entanglement" and Schrödinger's cat by Schrödinger and Einstein
- 1947 "Spooky action at a distance" in a letter to M. Born by A. Einstein
- 1976 Attempt to create quantum information theory
- 1980 Quantum mechanical model of Turing machine by Benioff (ANL)
- 1981 "Simulating Physics with Computers" by Feynman
- 1985 Quantum Turing machine by Deutsch
- 1992 Deutsch-Jozsa algorithm
- 1993 First paper on quantum teleportation
- 1994 Shor's factoring algorithm (cf RSA encryption)
- 1996 Grover search algorithm (Bell)
- 2004 First five photon entanglement by China
- 2011 First commercially available quantum computer (D-Wave)
- 2017 First quantum teleportation of independent single-photon qubit (14km) by China
- 2018 US National Quantum Initiative Act.
- 2019 Google quantum supremacy
- 2022 Nobel prize (Aspect, Clauser, Zeilinger) for violation of Bell's inequality
- 2022 433 qubits by IBM
- 2023 Breakthrough Prize (Bennet, Brassard, Shor, Deutsch)

Welcome to *quantum.gov*, the home of the National Quantum Initiative and its ongoing activities to explore and promote Quantum Information Science (QIS). The National Quantum Initiative Act provides for the continued leadership of the United States in QIS and its technology applications. It calls for a coordinated Federal program to accelerate quantum research and development for the economic and national security of the United States. The United States strategy for QIS R&D and related activities is described in the National Strategic Overview for QIS and supplementary documents.

LEARN MORE »

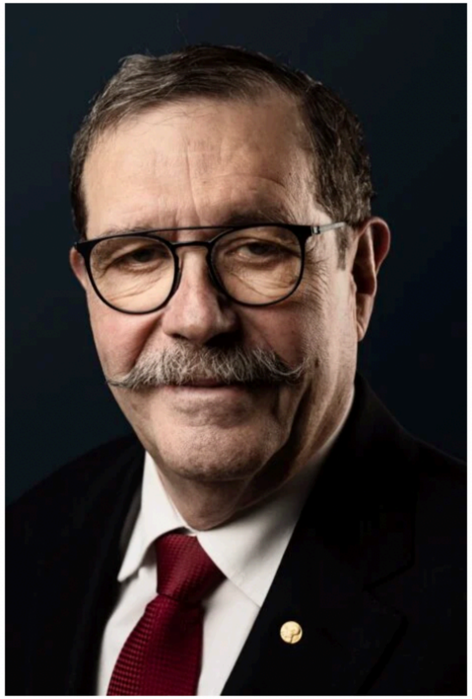
RECENT REPORTS

- Annual Report on the NQI Program Budget, January 6, 2023
- National Security Memorandum 10 on Quantum Computing, May 4, 2022
- Bringing Quantum Sensors to Fruition, March 24, 2022
- QIST Workforce Development National Strategic Plan, February 1, 2022
- The Role of International Talent in Quantum Information Science, October 5, 2021
- A Coordinated Approach to Quantum Networking Research, January 19, 2021
- Quantum Frontiers Report, October 7, 2020
- National Strategic Overview for Quantum Information Science, September 24, 2018

MORE PUBLICATIONS »

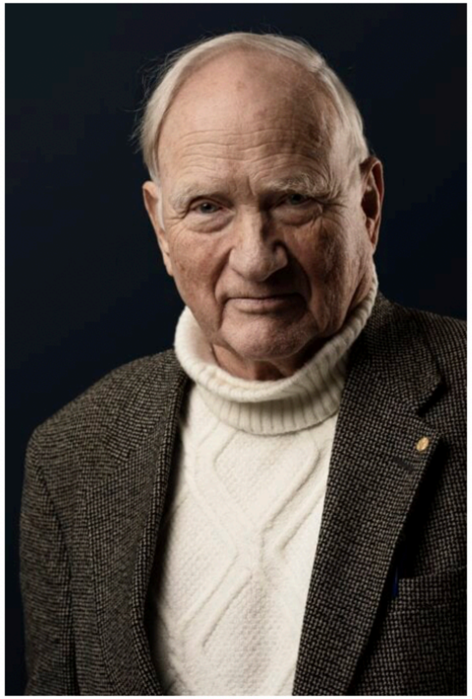
NATIONAL QUANTUM INITIATIVE 2018

The Nobel Prize in Physics 2022



Alain Aspect

Prize share: 1/3



John F. Clauser

Prize share: 1/3

Anton Zeilinger

Prize share: 1/3

The Nobel Prize in Physics 2022 was awarded jointly to Alain Aspect, John F. Clauser and Anton Zeilinger "for experiments with entangled photons, establishing the violation of Bell inequalities and pioneering quantum information science"

Yuri Milner found the Breakthrough prize in 2012.

FUNDAMENTAL PHYSICS
BREAKTHROUGH
PRIZE

BOARD TROPHY EVENTS NOMINATIONS NEWS CONTACTS
COMMITTEE PRIZES LAUREATES RULES
MANIFESTO

Search

LAUREATES

Breakthrough Prize   Special Breakthrough Prize   New Horizons Prize   Physics Frontiers Prize

2023   2022   2021   2020   2019   2018   2017   2016   2015   2014   2013   2012

Charles H. Bennett
IBM

Gilles Brassard

Université de Montréal

Oxford
David Deutsch

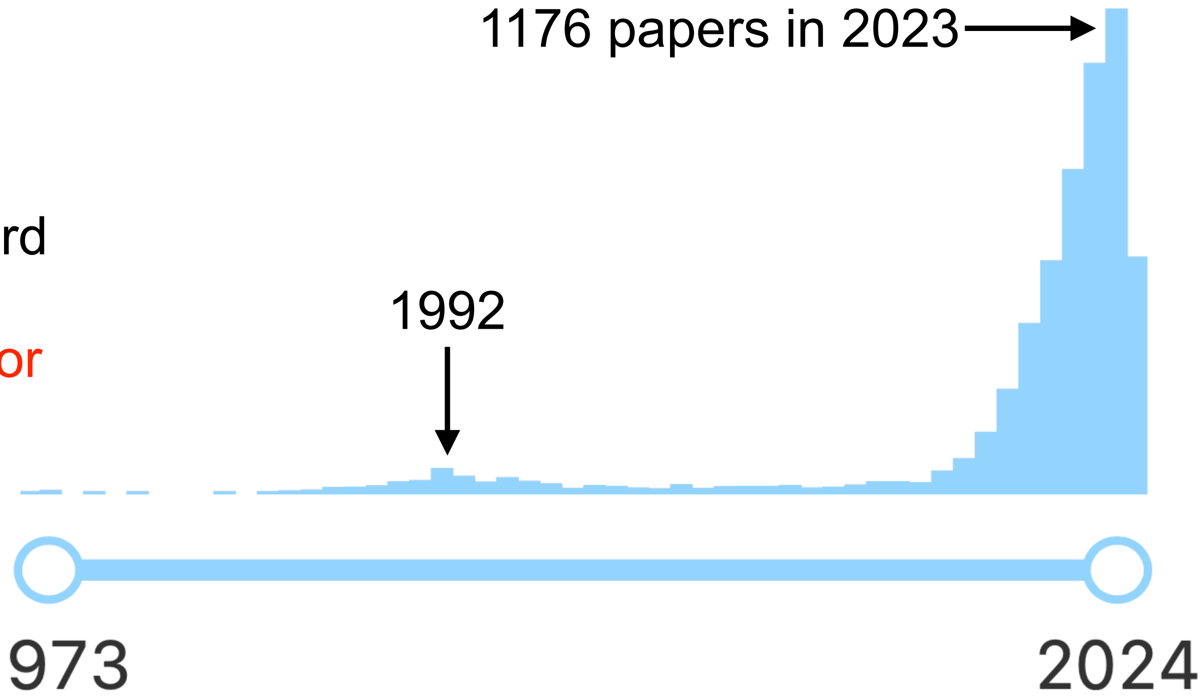
Peter W. Shor

MIT

BB84, quantum cryptography, factoring algorithms, Deutsch’s algorithm

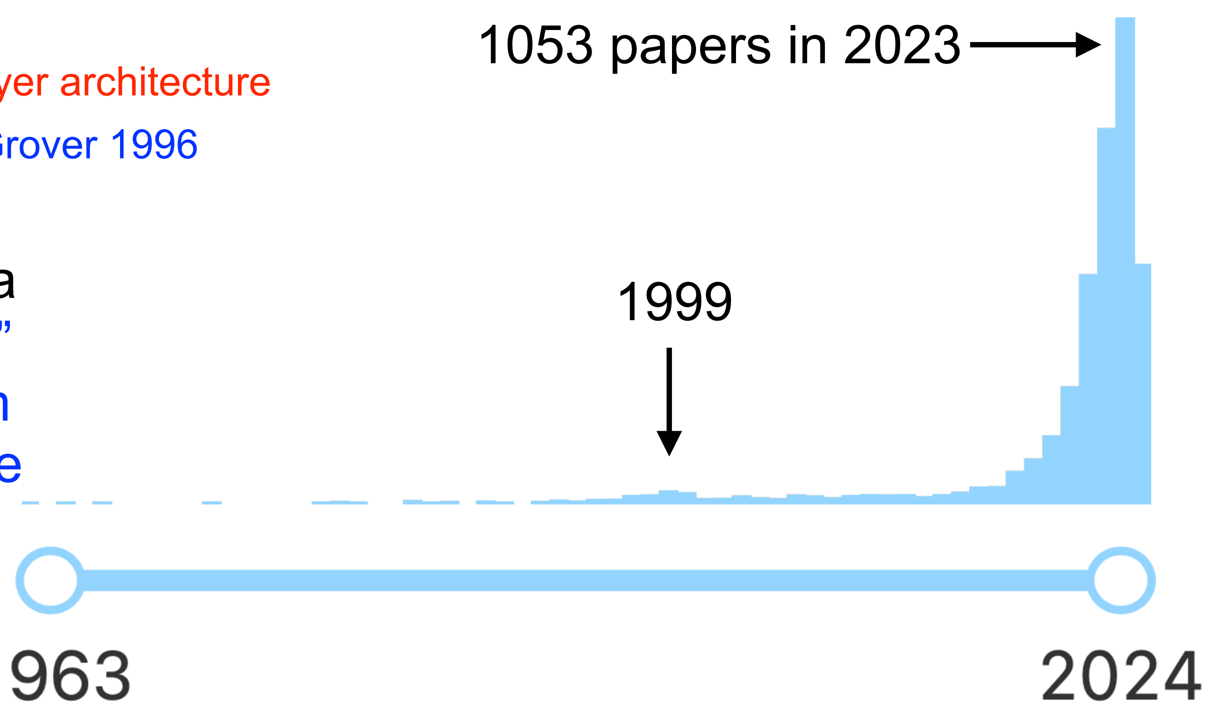
Data is obtained via **InspireHEP**

The number of papers (in high energy physics) that has a keyword “Machine Learning”, “Deep Learning”, “Artificial Intelligence” or “Neural Networks” in their title.



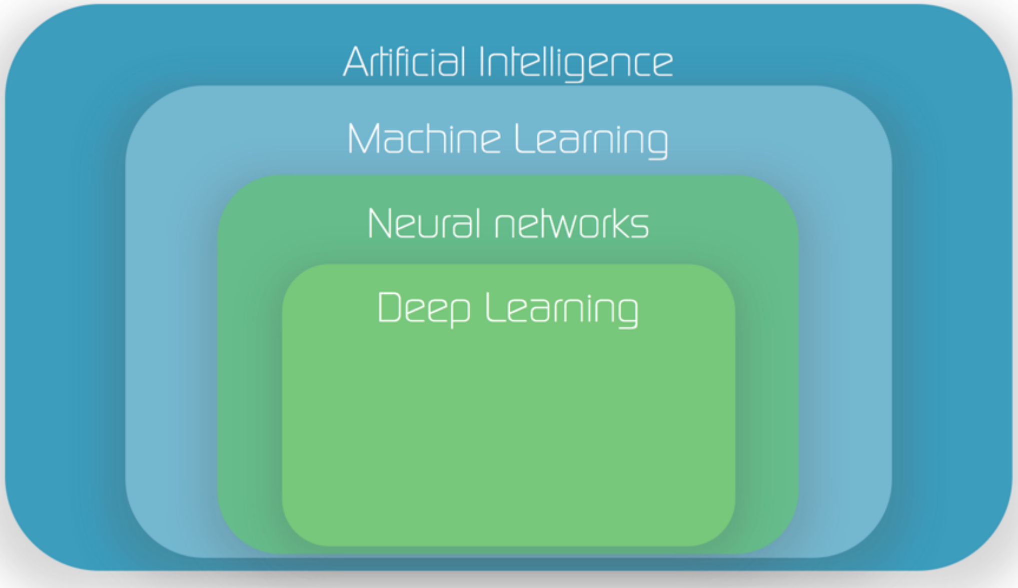
- G. Cybenko, 1989 with sigmoid activation
- K. Hornik, 1991, importance of the multilayer architecture
- D Simon, 1993, P. Shor 1994, 1995, L. Grover 1996

The number of papers that has a keyword “Quantum Computer”, ” Quantum Computing”,“Quantum Annealing” or “Quantum Machine Learning” in their title.



What is Machine Learning?

- Typically problems in physics can be formulated in terms of a search for some function $f : \mathbb{X} \to \mathbb{Y}$, from the space of the observed $\mathbb{X}$ to a low dimensional space of a desired target space/label $\mathbb{Y}$, which optimizes some metric (of our choice). The metric is often called a loss function and written as $L(\vec{y}, f(\vec{x}))$.
- A learning algorithm would find the function that optimizes L over all possible values of $(\vec{x}, \vec{y})$.
- But this is intractable owning to the curse of dimensionality and an infinite number of functions to choose from. Instead one has labeled training data $\{\vec{x}_i, \vec{y}_i\}_{i=1}^{N}$ sampled from $p(\vec{x}, \vec{y})$. Furthermore the function space is restricted to a model - a highly flexible family of functions $f_\phi(\vec{x})$ parameterized by ϕ .
- Sounds familiar?



Action Principle

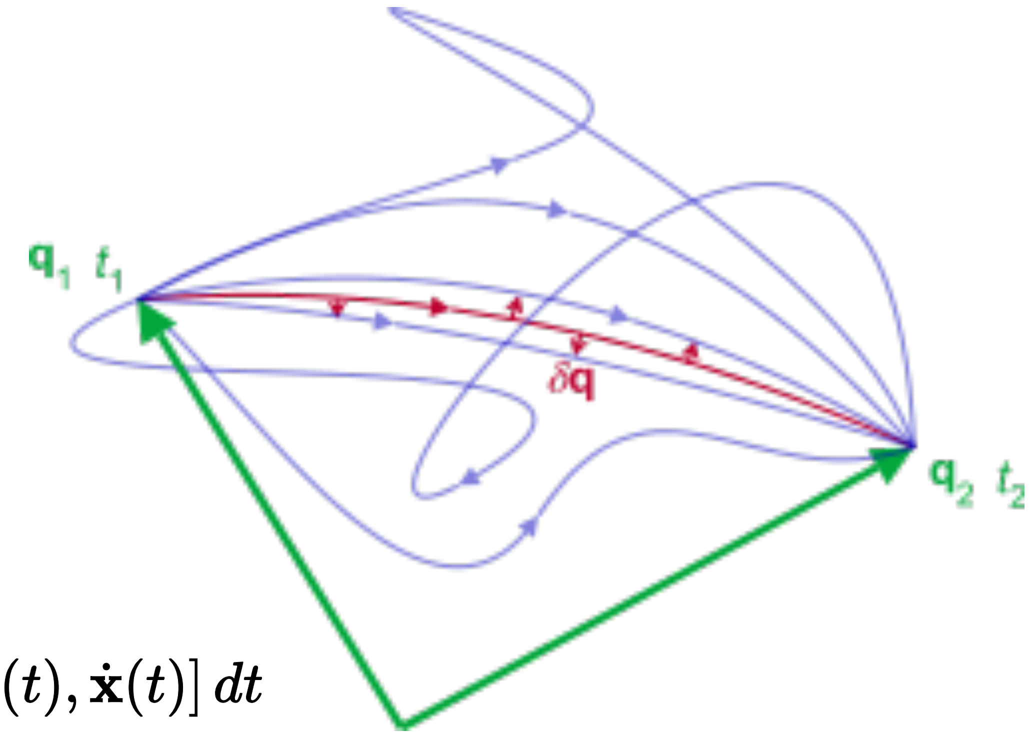
$$\mathbf{q} = (q_1,\ q_2,\ \dots\ ,\ q_N)$$

$$\mathbf{q} : \mathbf{R} \to \mathbf{R}^N$$

$$\mathcal{S}[\mathbf{q}, t_1, t_2] = \int_{t_1}^{t_2} L(\mathbf{q}(t), \dot{\mathbf{q}}(t), t)dt$$

$$\delta\mathcal{S} = 0$$

$$Z = \int e^{\frac{i\mathcal{S}[\mathbf{x}]}{\hbar}}\, \mathcal{D}\mathbf{x} \quad \text{where } \mathcal{S}[\mathbf{x}] = \int_0^{t_f} L[\mathbf{x}(t), \dot{\mathbf{x}}(t)]\, dt$$



$$Z \sim \int e^{-S}\ Dx$$

- Newton’s equation of motion
- Maxwell’s equation
- Schrodinger / Dirac equation
- General relativity
- All other fundamental equation of motions

Universal Approximation Theorem

- A feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of R^n, under mild assumptions on the activation function.

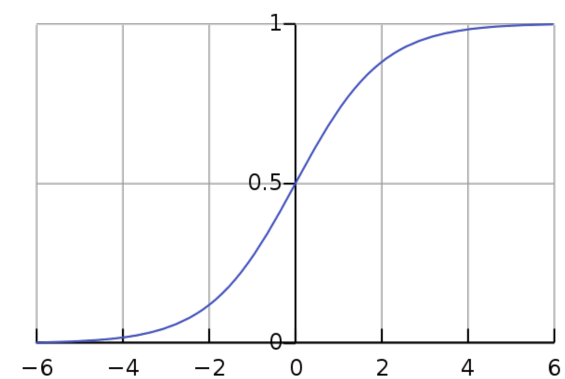
Let $\varphi : \mathbb{R} \to \mathbb{R}$ be a nonconstant, bounded, and continuous function (called the *activation function*). Let I_m denote the m-dimensional unit hypercube $[0,1]^m$. The space of real-valued continuous functions on I_m is denoted by $C(I_m)$. Then, given any $\varepsilon > 0$ and any function $f \in C(I_m)$, there exist an integer N, real constants $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$ for $i = 1, \ldots, N$, such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i \varphi\left(w_i^T x + b_i\right)$$ as an approximate realization of the function f; that is,

$$|F(x) - f(x)| < \varepsilon$$ for all $x \in I_m$. In other words, functions of the form $F(x)$ are dense in $C(I_m)$.

- A. N. Kolmogorov, 1957
- G. Cybenko, 1989 with sigmoid activation
- K. Hornik, 1991, importance of the multilayer architecture
- Z. Lu et al, 2017, with deep neural network and ReLu activation

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$



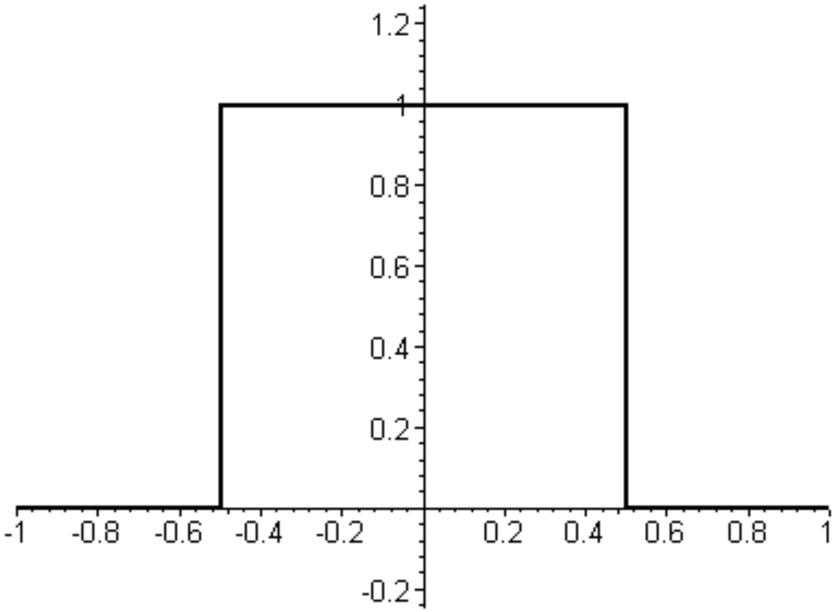
Let $p > 0$ be a fixed number and $f(x)$ be a periodic function with period $2p$, defined on $(-p, p)$. The Fourier series of $f(x)$ is a way of expanding the function $f(x)$ into an infinite series involving sines and cosines:

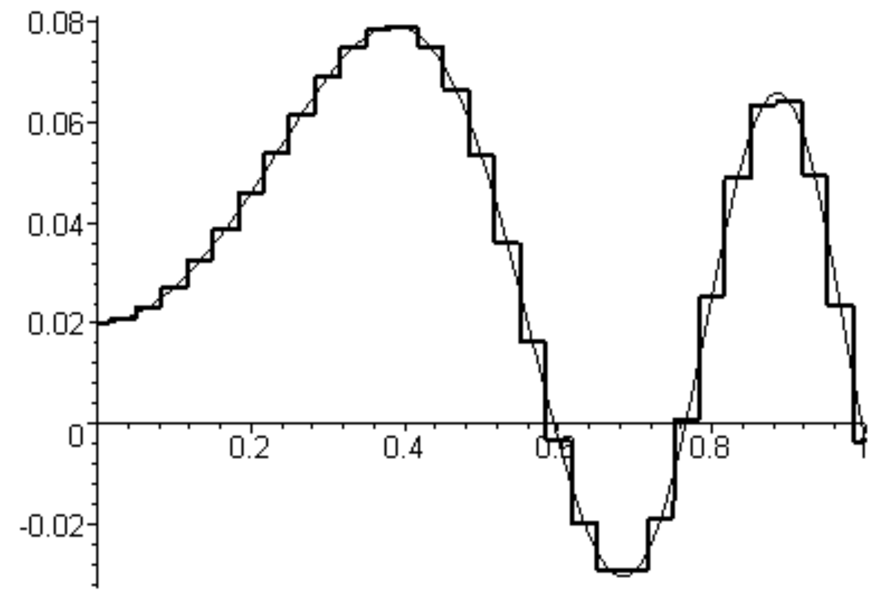
$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\frac{n\pi x}{p}) + \sum_{n=1}^{\infty} b_n \sin(\frac{n\pi x}{p}) \tag{2.1}$$

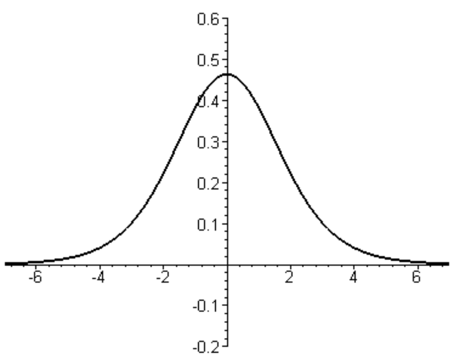
where a_0, a_n, and b_n are called the Fourier coefficients of $f(x)$, and are given by the formulas

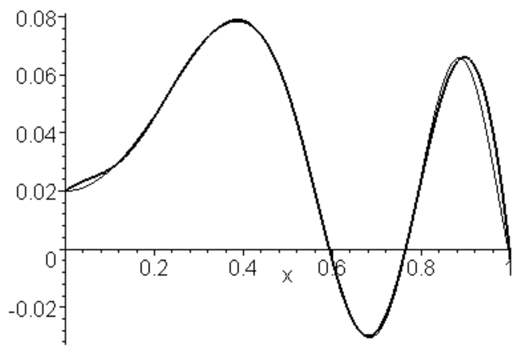
$$a_0 = \frac{1}{p}\int_{-p}^{p} f(x)\, dx, \qquad a_n = \frac{1}{p}\int_{-p}^{p} f(x)\cos(\frac{n\pi x}{p})\, dx, \tag{2.2}$$

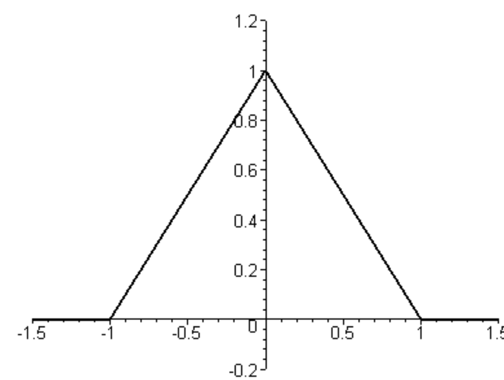
$$b_n = \frac{1}{p}\int_{-p}^{p} f(x)\sin(\frac{n\pi x}{p})\, dx,$$

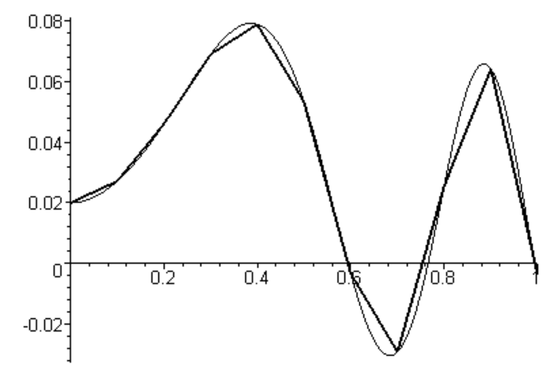

1.2
1
0.8
0.6
0.4
0.2
-0.2
-1 -0.8 -0.6 -0.4 -0.2 0.2 0.4 0.6 0.8 1


0.08
0.06
0.04
0.02
0
-0.02
0.2 0.4 0.8 1


0.6
0.5
0.4
0.3
0.2
0.1
-0.1
-0.2
-6 -4 -2 2 4 6

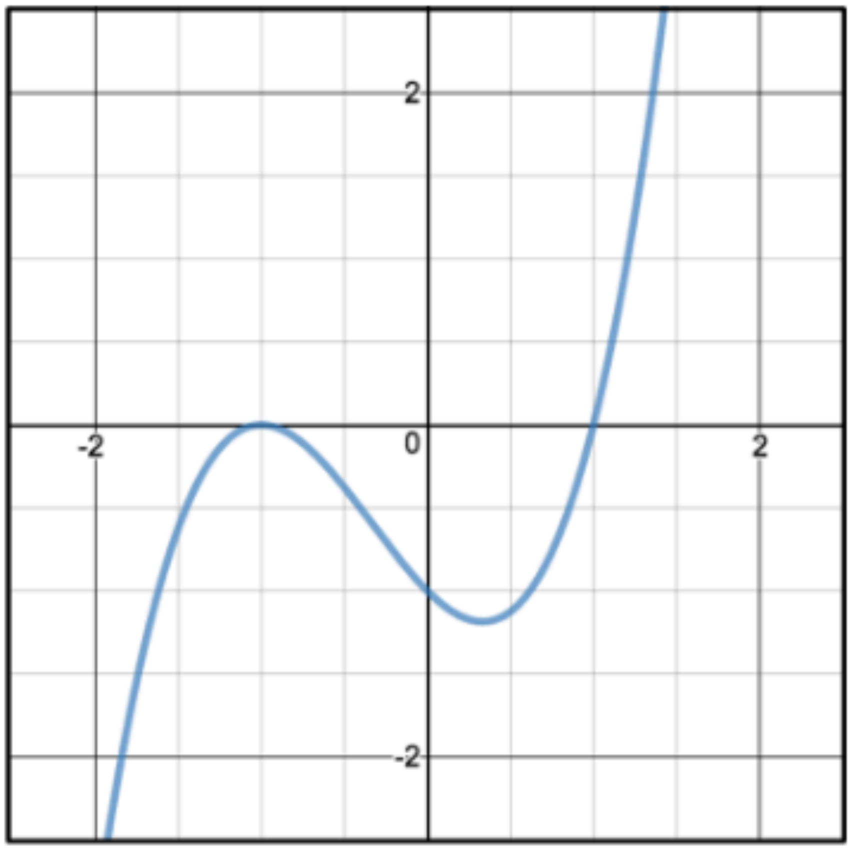

0.08
0.06
0.04
0.02
0
-0.02
0.2 0.4 x 0.6 0.8 1


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-1.5 -1 -0.5 0.5 1 1.5


0.08
0.06
0.04
0.02
0
-0.02
0.2 0.4 0.6 0.8 1

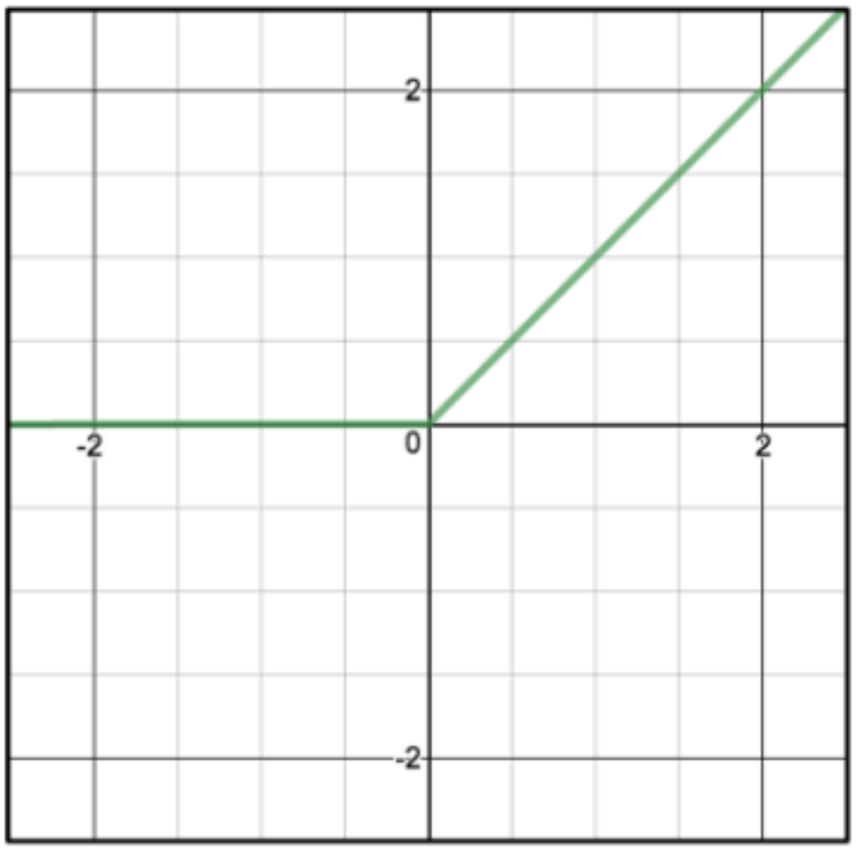
Neural network is a function-approximator.

$f(x) = x^3 + x^2 - x - 1$



Rectified Linear Unit

ReLU



ReLu = max(0, x)

Example by Joe Klein

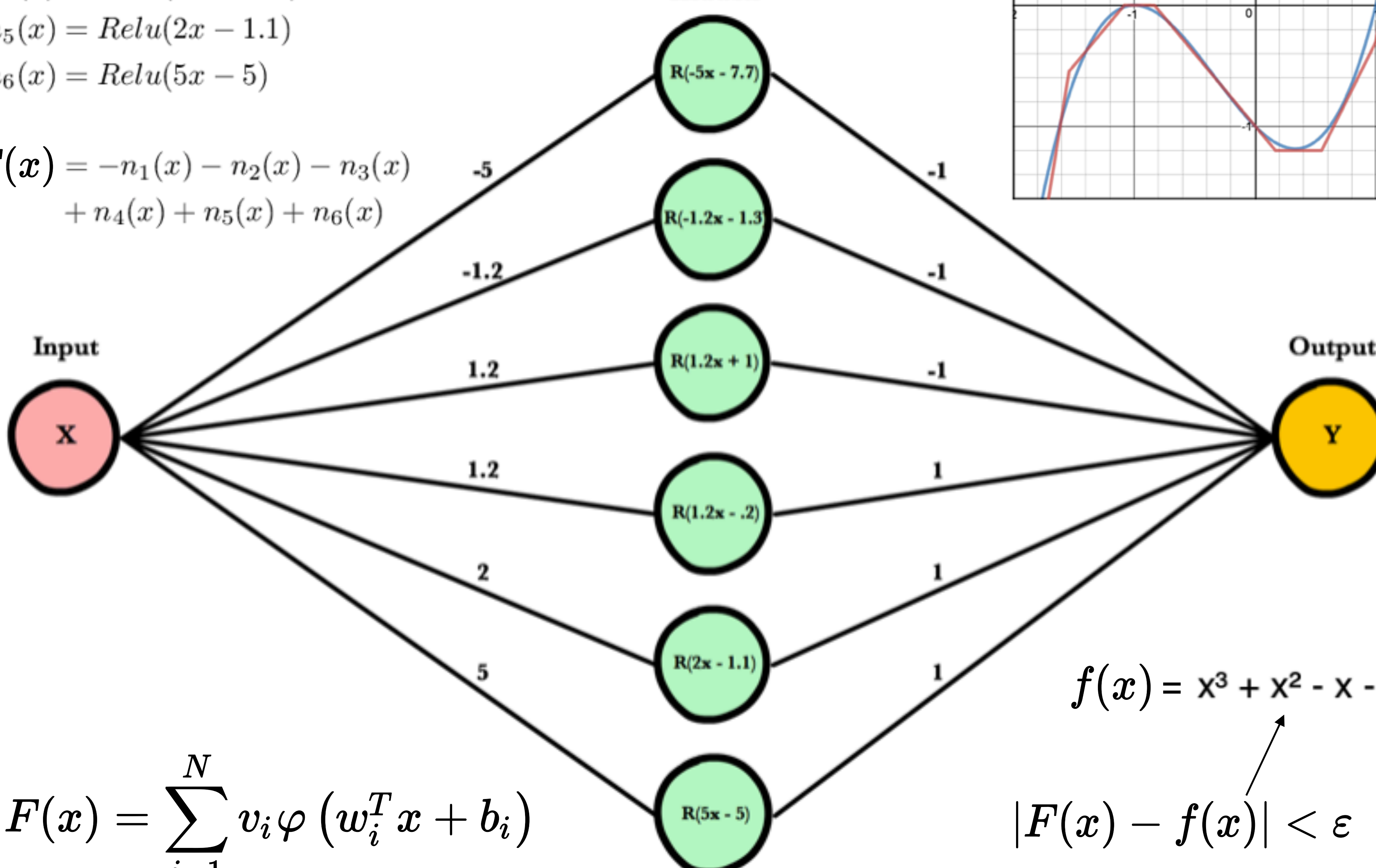
$f(x) = x^3 + x^2 - x - 1$

$$n_1(x) = Relu(-5x - 7.7)$$
$$n_2(x) = Relu(-1.2x - 1.3)$$
$$n_3(x) = Relu(1.2x + 1)$$
$$n_4(x) = Relu(1.2x - .2)$$
$$n_5(x) = Relu(2x - 1.1)$$
$$n_6(x) = Relu(5x - 5)$$

$$F(x) = -n_1(x) - n_2(x) - n_3(x) + n_4(x) + n_5(x) + n_6(x)$$

$$n_1(x) = Relu(-5x - 7.7)$$
$$n_2(x) = Relu(-1.2x - 1.3)$$
$$n_3(x) = Relu(1.2x + 1)$$
$$n_4(x) = Relu(1.2x - .2)$$
$$n_5(x) = Relu(2x - 1.1)$$
$$n_6(x) = Relu(5x - 5)$$

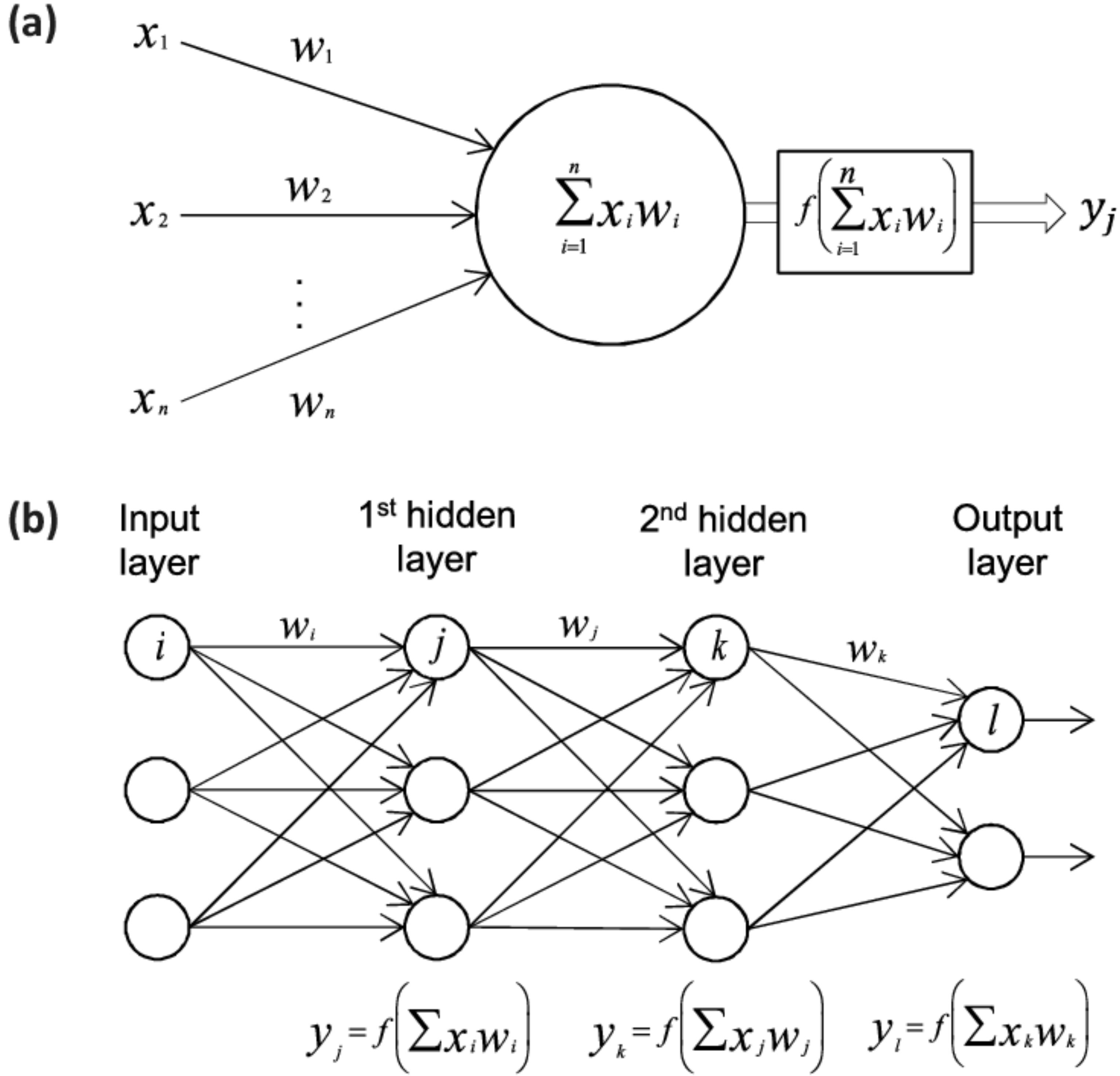
$$F(x) = -n_1(x) - n_2(x) - n_3(x) + n_4(x) + n_5(x) + n_6(x)$$



$$f(x) = x^3 + x^2 - x - 1$$

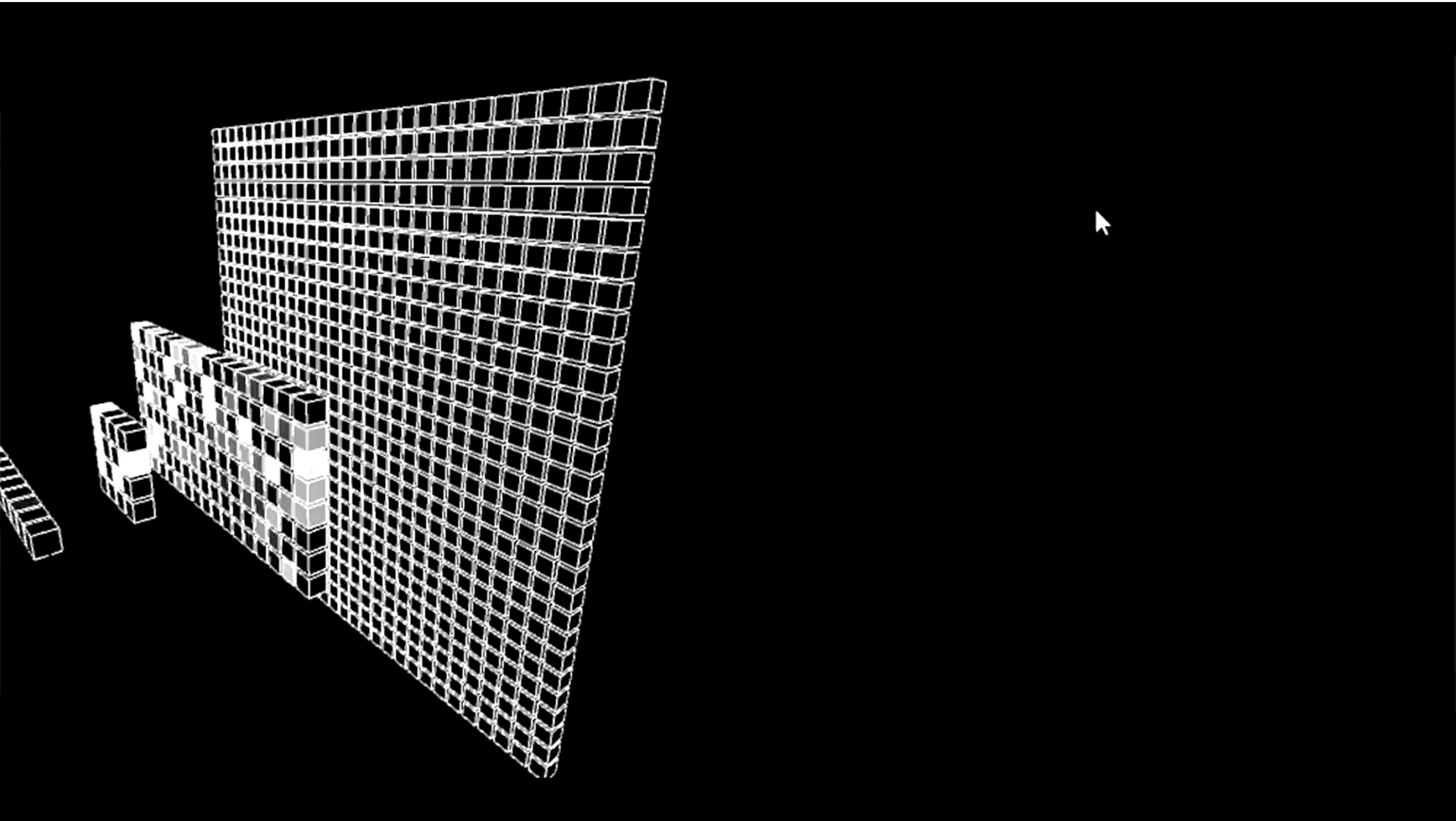
$$|F(x) - f(x)| < \varepsilon$$

$$F(x) = \sum_{i=1}^{N} v_i \varphi\left(w_i^T x + b_i\right)$$


(a)
x_1
w_1
x_2
w_2
x_n
w_n
$\sum_{i=1}^{n} x_i w_i$
$f\left(\sum_{i=1}^{n} x_i w_i\right)$
y_j
(b)
Input layer
1st hidden layer
2nd hidden layer
Output layer
i
j
k
l
w_i
w_j
w_k
$y_j = f\left(\sum x_i w_i\right)$
$y_k = f\left(\sum x_j w_j\right)$
$y_l = f\left(\sum x_k w_k\right)$

Object identification

Taken from Vecanoi (Youtube Educational channel about AI)



This site quizzes 17 Verbal & 8 Vision AIs every week | Last Updated: 06:14PM EDT on June 14, 2025

About Offline Test ⓘ *About Mensa Norway* ⓘ

IQ Test Results

Score reflects average of last 7 tests given

Reset | Show Offline Test | Show Mensa Norway

Vision Only | Vision Only

TrackingAI.org

Offline quiz (not in training data)

A MaximumTruth.org Project

Why Machine Learning?

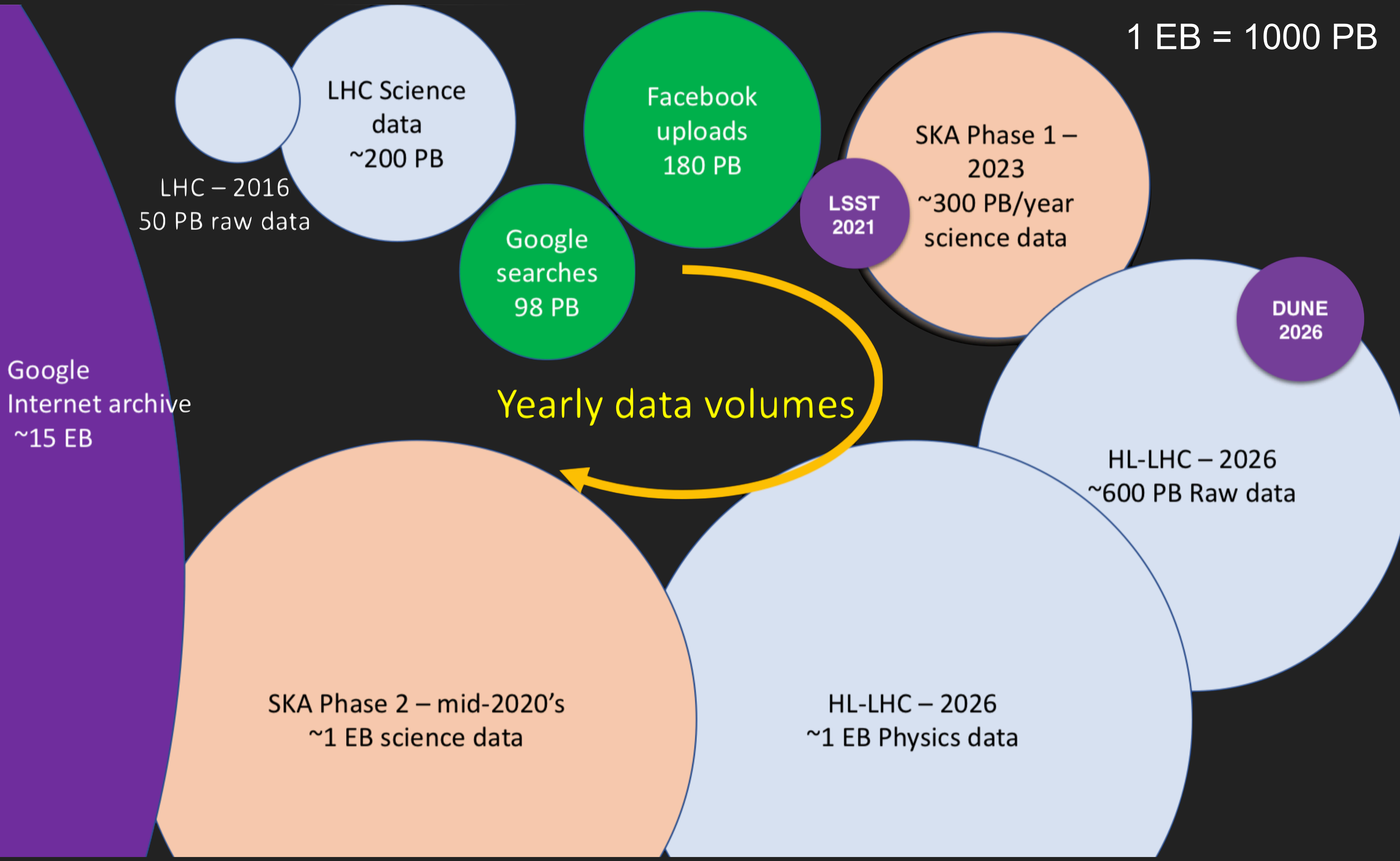
CHALLENGE: BIG DATA

Taken from J. Duarte's talk

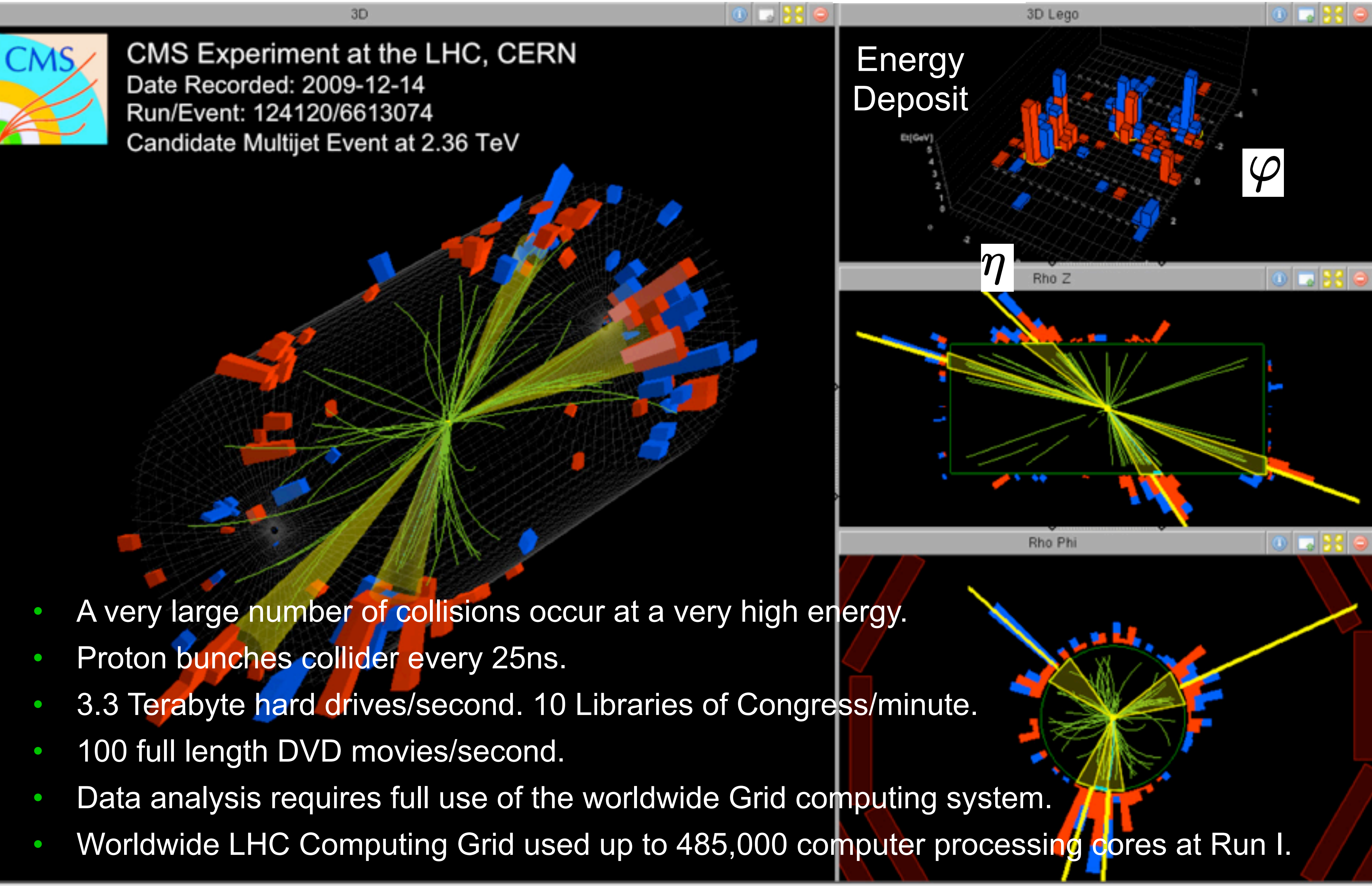
- HL-LHC will reach 1 exabyte of data per year

1 PB = 1000 TB

1 EB = 1000 PB



$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$



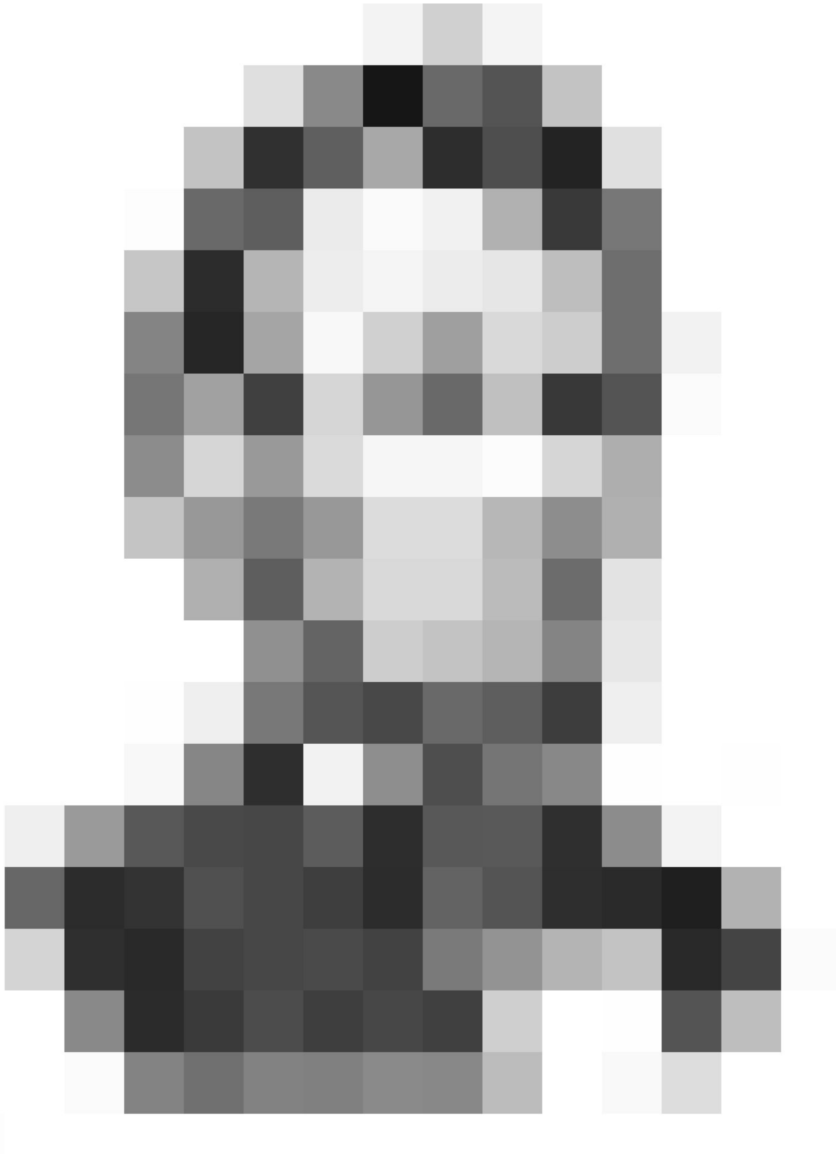
- A very large number of collisions occur at a very high energy.
- Proton bunches collider every 25ns.
- 3.3 Terabyte hard drives/second. 10 Libraries of Congress/minute.
- 100 full length DVD movies/second.
- Data analysis requires full use of the worldwide Grid computing system.
- Worldwide LHC Computing Grid used up to 485,000 computer processing cores at Run I.

- Perception (our understanding of the universe) is a dynamic combination of top-down (theory) and bottom-up (data driven) processing.

I. Shipsey, 1707.03711

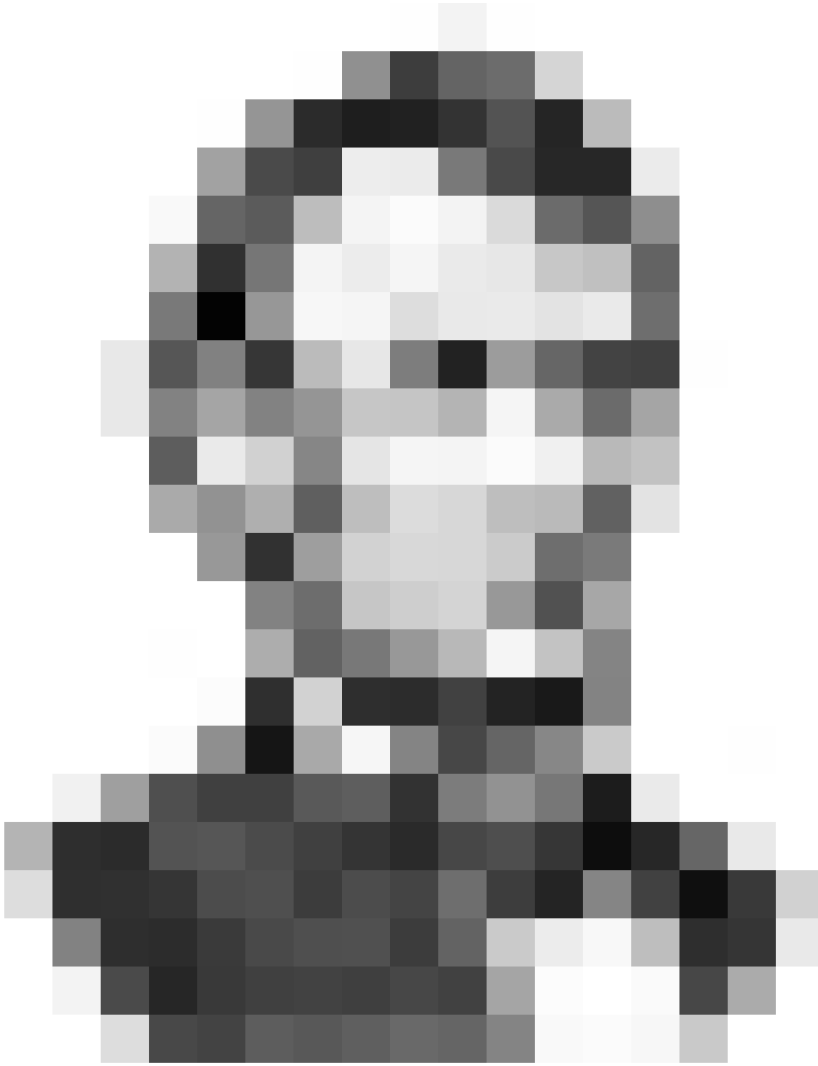
TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]



TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]



TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

Abraham Lincoln

TOP-DOWN AND BOTTOM-UP

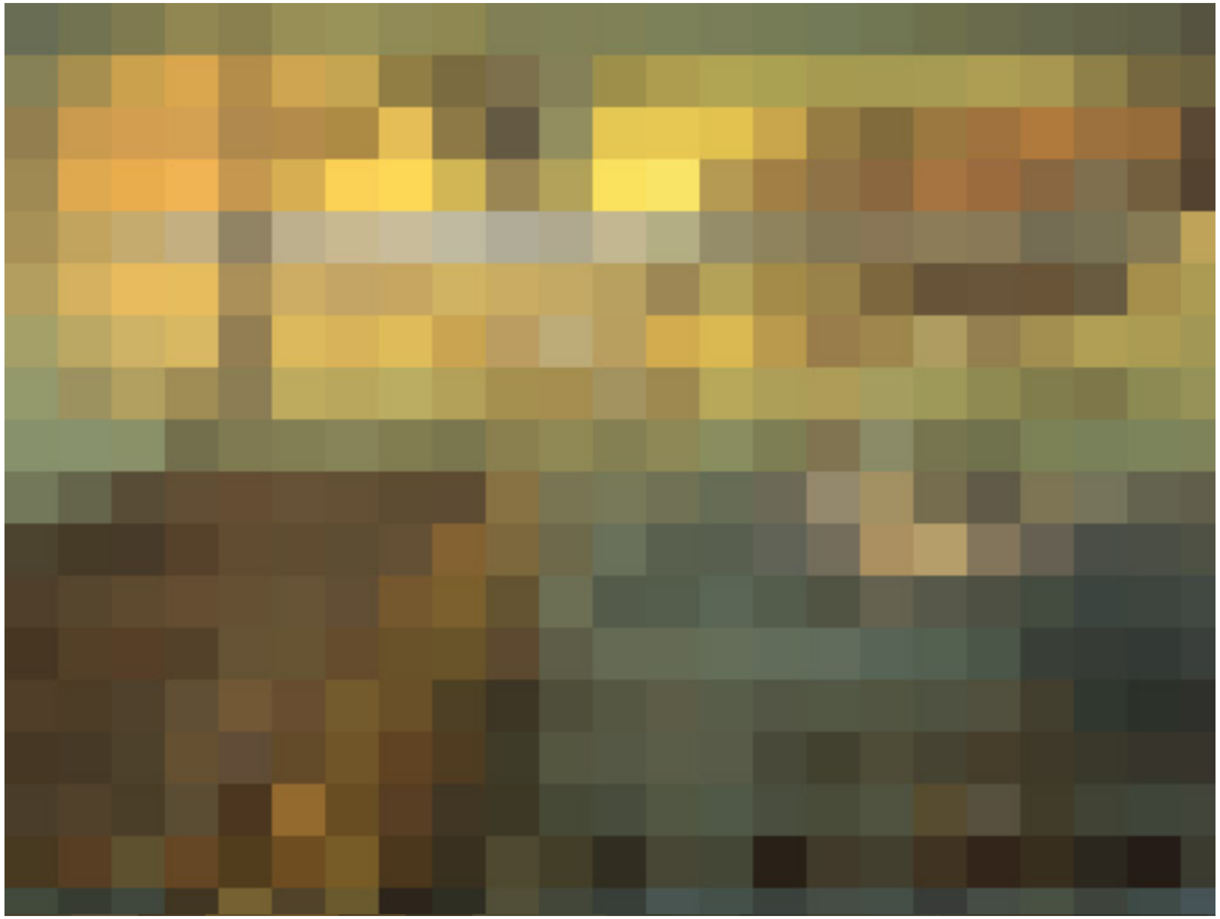
Credit: [Ian Shipsey]

Abraham Lincoln

TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

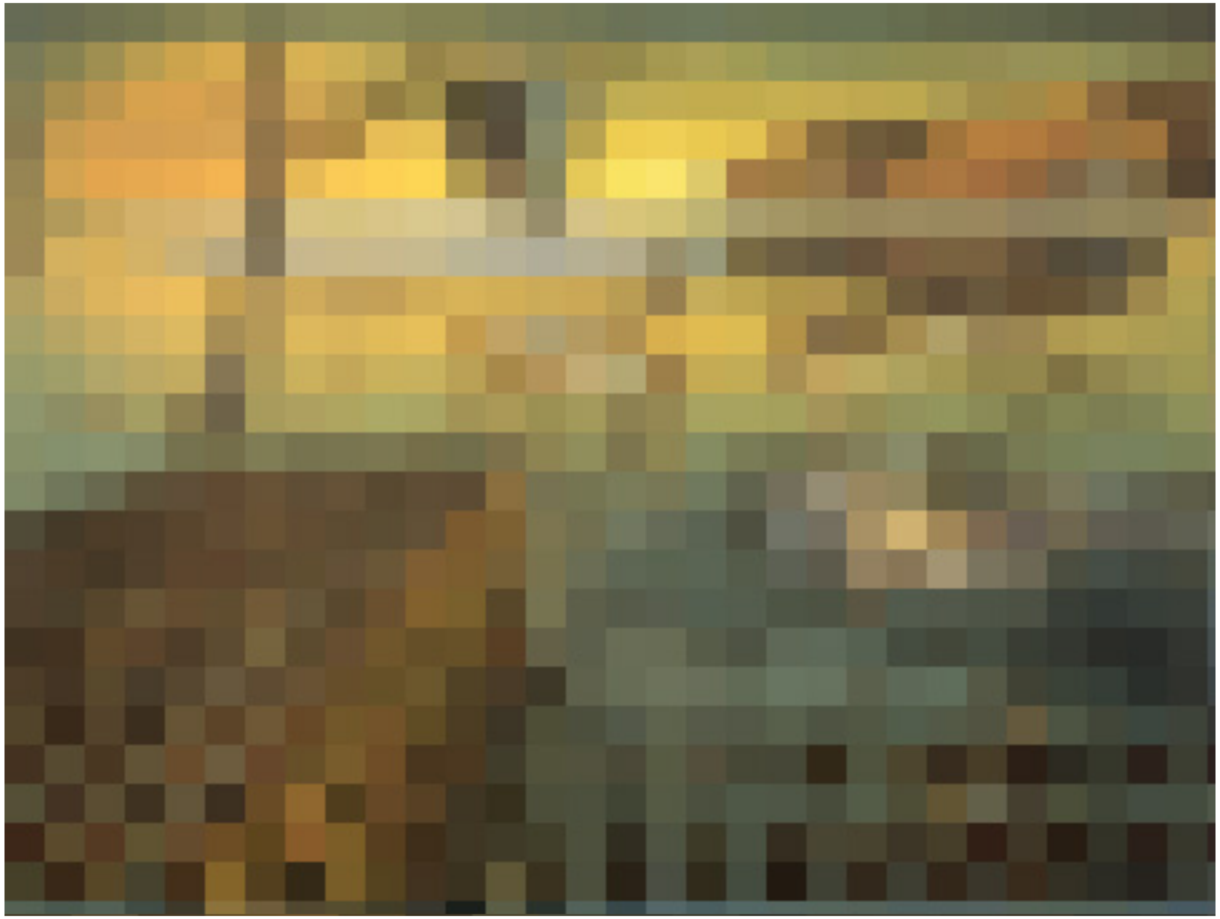
Abraham Lincoln



TOP-DOWN AND BOTTOM-UP

Credit: Ian Shipsey

Abraham Lincoln



TOP-DOWN AND BOTTOM-UP

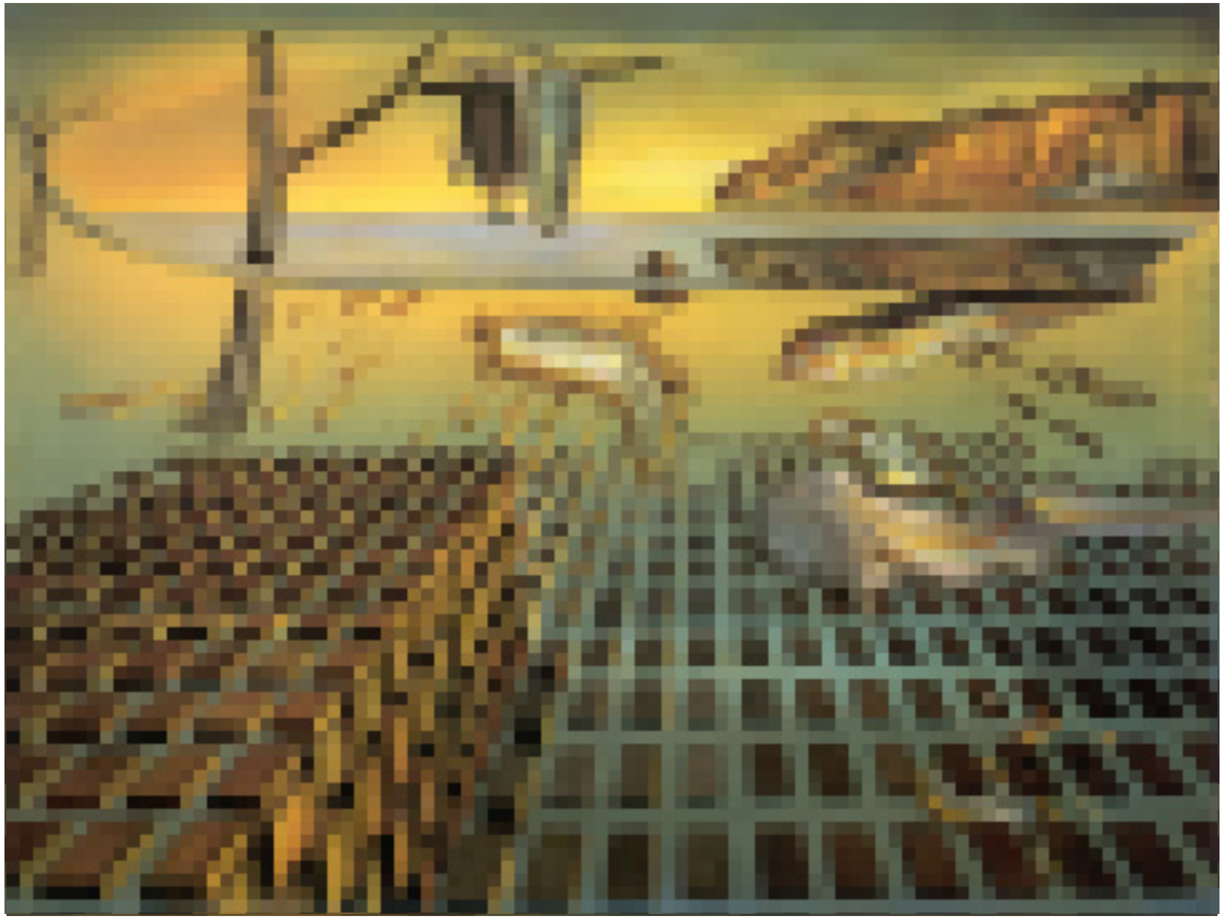
Credit: [Ian Shipsey]

Abraham Lincoln

TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

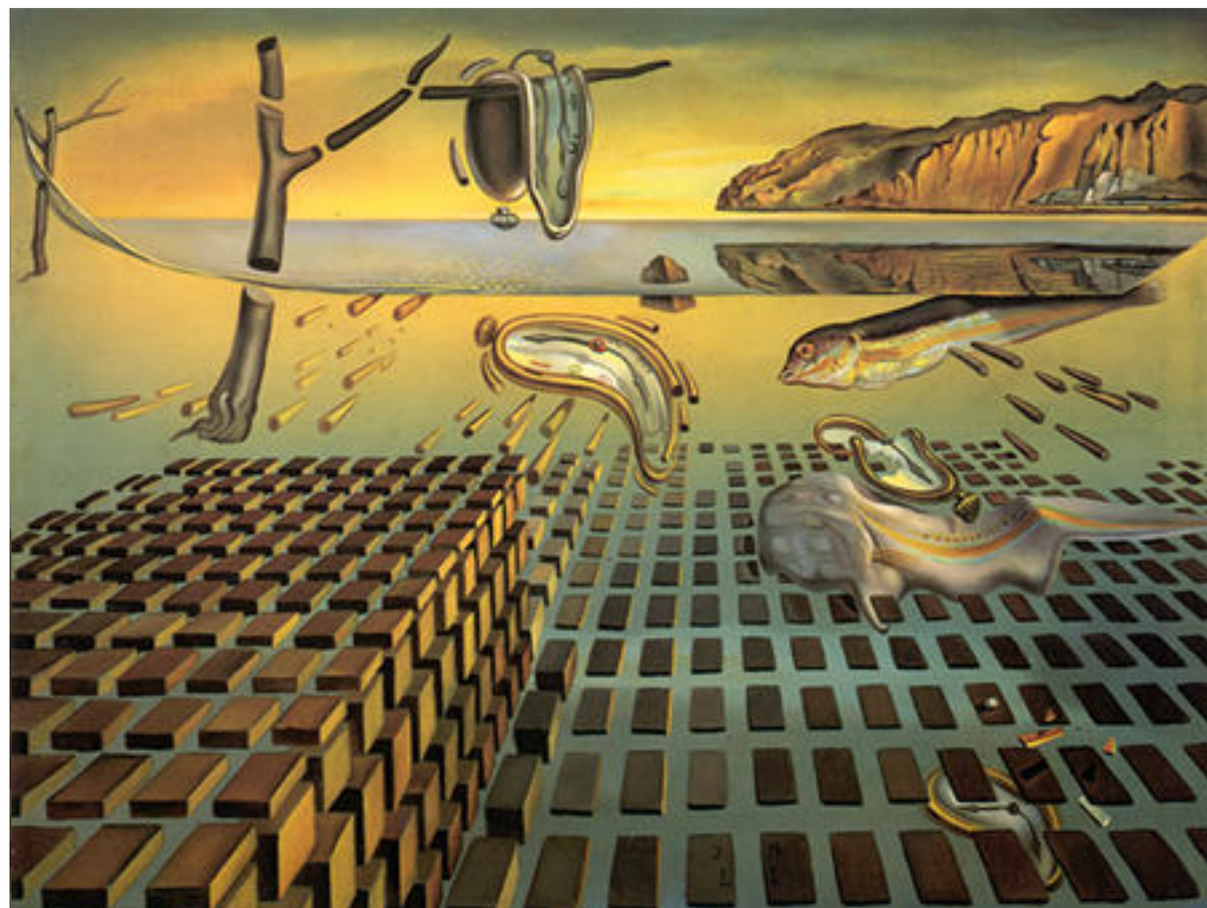
Abraham Lincoln



TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

Abraham Lincoln



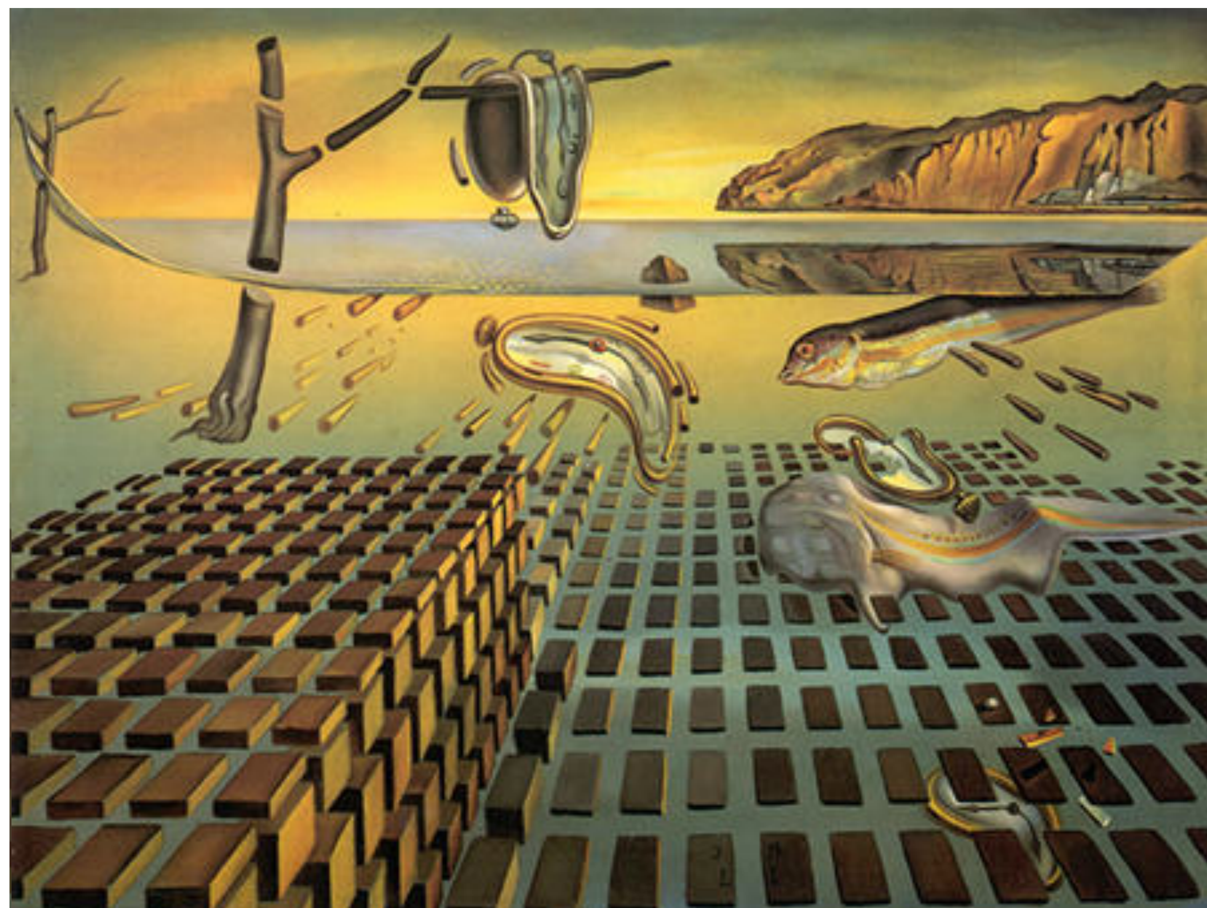
TOP-DOWN AND BOTTOM-UP

Credit: [Ian Shipsey]

1707.03711

Abraham Lincoln

Salvador Dali's The Disintegration of the Persistence of Memory



- Perception (our understanding of the universe) is a dynamic combination of top-down (theory) and bottom-up (data driven) processing.
- From 1967 to 2012 particle physics was in a situation very similar to recognizing the image of Lincoln. Since 2012 we are in a situation where we are trying to recognize a Dali masterpiece, with little information to guide us. Without a roadmap we are dependent on bottom up information: we are in a data driven era. (Theoretical challenges)
- Machine Learning and Quantum Computing are valuable tools for new physics searches.

I. Shipsey, 1707.03711

What is quantum computing?

Digital Computing vs Quantum Computing

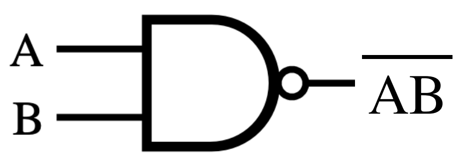
Use binaries to perform calculations and solve problems

Bit: $x \in \{0,1\}$

Digital computation with n bits:

$$\{0,1\}^n \longrightarrow \{0,1\}^m, \quad m \leq n$$

Classical gates:



NAND:

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

Algorithms

Use QM / qubits to solve problems

Qubit: $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle = \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{pmatrix}$

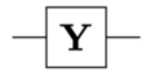
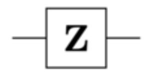
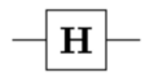
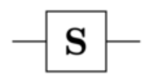
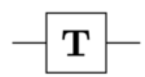
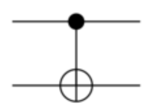
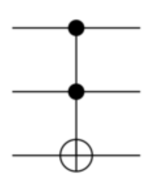
Quantum computation with n qubits: $\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{pmatrix} \in \mathbb{C}^{2^n}$

Quantum gates: Unitary transformation
Measurement

Rotation around x-axis: $R(\theta) = \exp\left(-i\theta\frac{\sigma_x}{2}\right)$

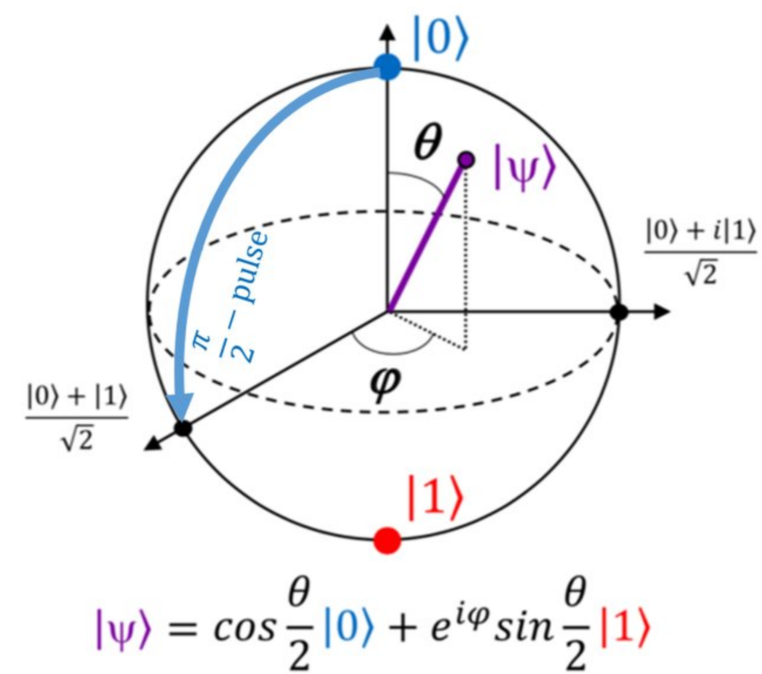
Pauli X: $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Quantum Algorithms

Operator	Gate(s)	Matrix
Pauli-X (X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Why Quantum Computing?

- Cryptography
  - Mathematics: factoring, hidden subgroup program, discrete logarithm problem
- Optimization
- Search algorithm
- Quantum Machine Learning
  - Quantum Advantages?
    - Learns better with small # of data
    - Faster convergence
    - Less # of parameters
- Quantum simulation
- What are the interesting problems?



The First Wave of Quantum Machine Learning?

PRL **103,** 150502 (2009) PHYSICAL REVIEW LETTERS week ending 9 OCTOBER 2009

Quantum Algorithm for Linear Systems of Equations

Aram W. Harrow,[1] Avinatan Hassidim,[2] and Seth Lloyd[3]

[1]*Department of Mathematics, University of Bristol, Bristol, BS8 1TW, United Kingdom*
[2]*Research Laboratory for Electronics, MIT, Cambridge, Massachusetts 02139, USA*
[3]*Research Laboratory for Electronics and Department of Mechanical Engineering, MIT, Cambridge, Massachusetts 02139, USA*

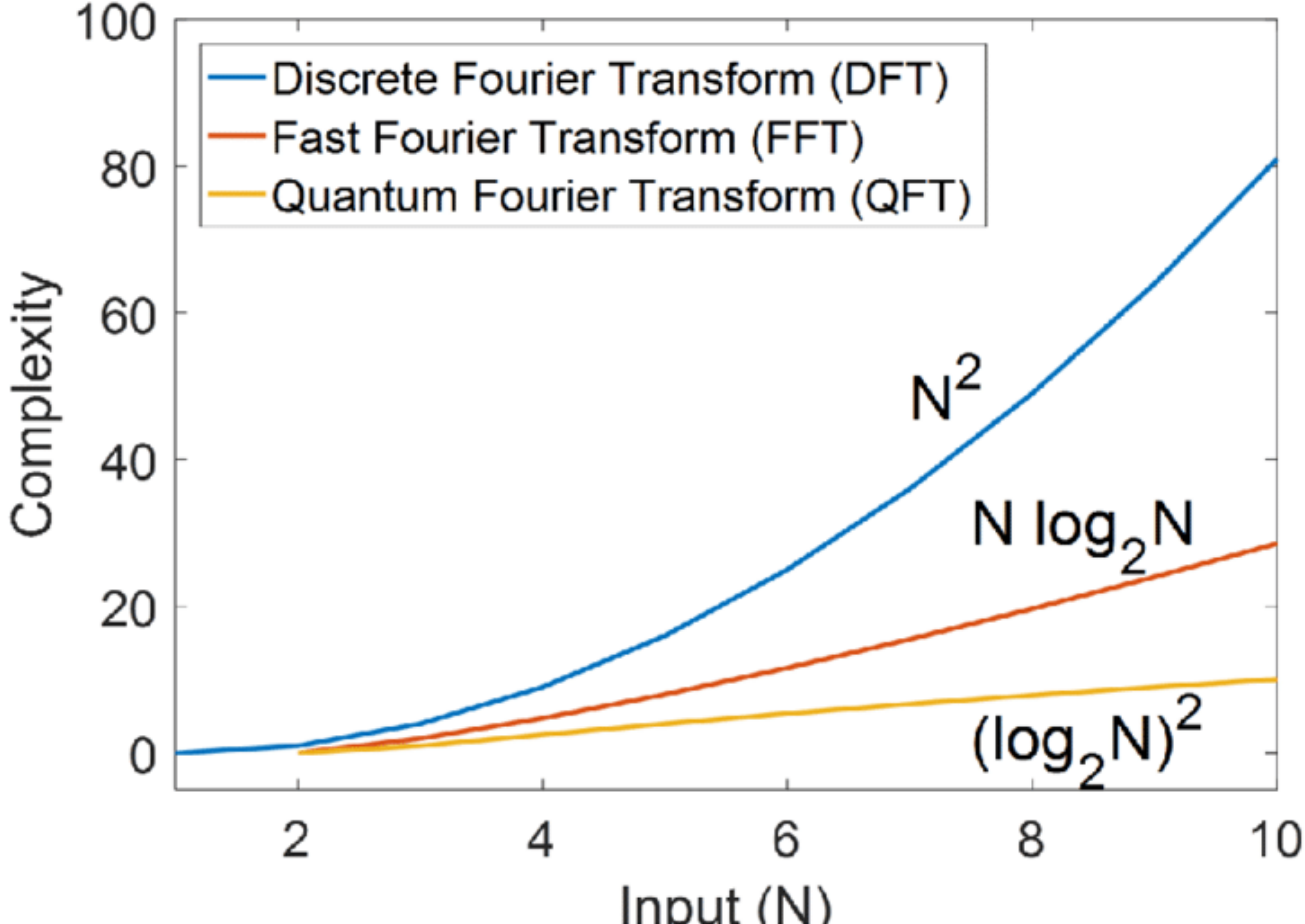
(Received 5 July 2009; published 7 October 2009)

Solving linear systems of equations is a common problem that arises both on its own and as a subroutine in more complex problems: given a matrix A and a vector $\vec{b}$, find a vector $\vec{x}$ such that $A\vec{x} = \vec{b}$. We consider the case where one does not need to know the solution $\vec{x}$ itself, but rather an approximation of the expectation value of some operator associated with $\vec{x}$, e.g., $\vec{x}^\dagger M\vec{x}$ for some matrix M. In this case, when A is sparse, $N \times N$ and has condition number κ, the fastest known classical algorithms can find $\vec{x}$ and estimate $\vec{x}^\dagger M\vec{x}$ in time scaling roughly as $N\sqrt{\kappa}$. Here, we exhibit a quantum algorithm for estimating $\vec{x}^\dagger M\vec{x}$ whose runtime is a polynomial of $\log(N)$ and κ. Indeed, for small values of κ [i.e., poly $\log(N)$], we prove (using some common complexity-theoretic assumptions) that any classical algorithm for this problem generically requires exponentially more time than our quantum algorithm.

$$Ax = b$$

Complexity of inversion of a regular matrix=$O(N^3)$

Complexity of inversion of a sparse matrix=$O(N)$



Discrete Fourier Transform (DFT)
Fast Fourier Transform (FFT)
Quantum Fourier Transform (QFT)
N^2
$N \log_2 N$
$(\log_2 N)^2$
Complexity
Input (N)
0
20
40
60
80
100
2
4
6
8
10

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31
33	35	37	39	41	43	45	47
49	51	53	55	57	59	61	63

2	3	6	7	10	11	14	15
18	19	22	23	26	27	30	31
34	35	38	39	42	43	46	47
50	51	54	55	58	59	62	63

4	5	6	7	12	13	14	15
20	21	22	23	28	29	30	31
36	37	38	39	44	45	46	47
52	53	54	55	60	61	62	63

8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31
40	41	42	43	44	45	46	47
56	57	58	59	60	61	62	63

16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

XXXXX1

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31
33	35	37	39	41	43	45	47
49	51	53	55	57	59	61	63

XXXX1X

2	3	6	7	10	11	14	15
18	19	22	23	26	27	30	31
34	35	38	39	42	43	46	47
50	51	54	55	58	59	62	63

XXX1XX

4	5	6	7	12	13	14	15
20	21	22	23	28	29	30	31
36	37	38	39	44	45	46	47
52	53	54	55	60	61	62	63

8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31
40	41	42	43	44	45	46	47
56	57	58	59	60	61	62	63

XX1XXX

16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

X1XXXX

32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

1XXXXX

Is there a way to find out the answer by asking less than 6 questions?

Bernstein-Vazirani Algorithm

- A n-bit function $f: \{0,1\}^{\otimes n} \longrightarrow \{0,1\}$, which outputs a single bit, is guaranteed to be of the form $f_s(x) = x \cdot s$, where s is an unknown n-bit string and $x \cdot s = x_0 s_0 + \cdots + x_{n-1} s_{n-1} = \sum_{i=0}^{n-1} x_i s_i \pmod 2$. Find the unknown string $s = (s_0 s_1 \cdots s_{n-1})$.
- Best classical algorithm uses $\mathcal{O}(n)$ calls to $f_s(x) = x \cdot s \mod 2$. Each query gives one bit of information of s (because f outputs one bit).

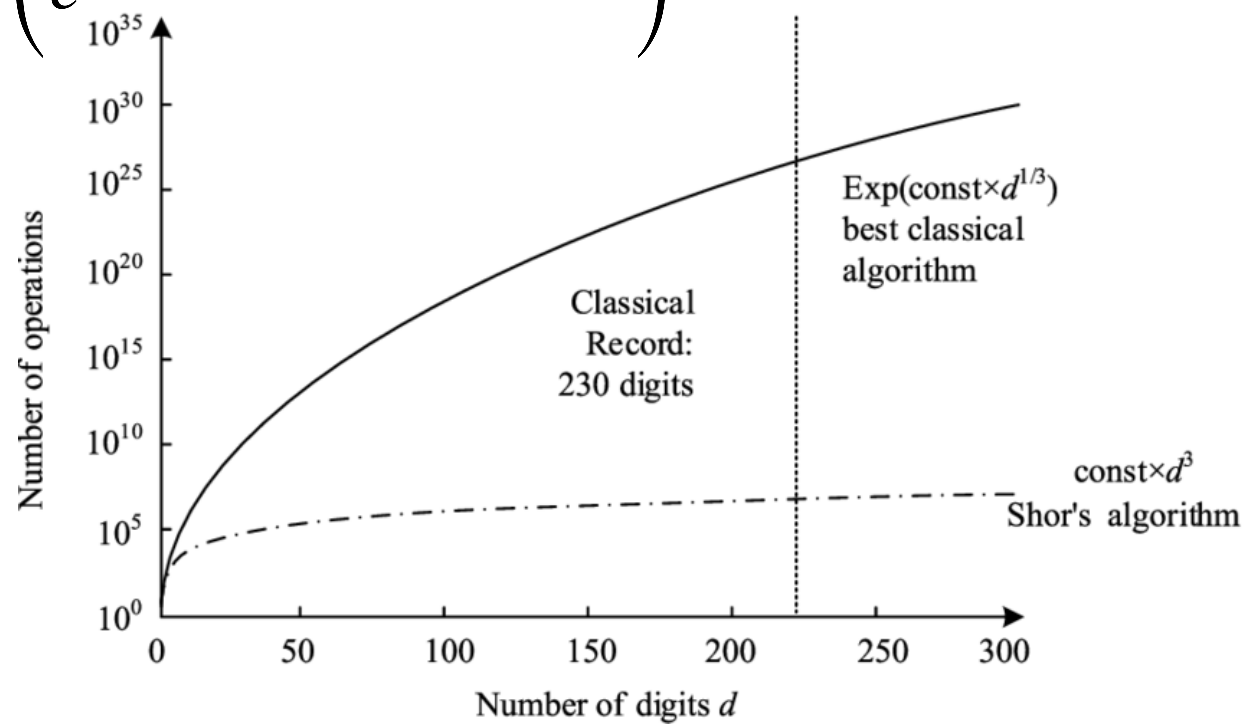
- How do we find s with less than n queries? → Use superposition (over all possible input bit strings)

Shor's factoring algorithm

- To factor an integer N, Shor's algorithm runs in polynomial time, meaning the time taken is polynomial in $\log N$, the size of the integer given as input. Specifically, it takes quantum gates of order $O\left((\log N)^2(\log\log N)(\log\log\log N)\right)$.
- This is significantly faster than the most efficient known classical factoring algorithm, the general number field sieve, which works in sub-exponential time: $O\left(e^{1.9(\log N)^{1/3}(\log\log N)^{2/3}}\right)$

```
RSA-250 =
6413528947707158027879019017057738908482501474294344720811685963202453234463023862359875266834770873766192558569463979885 3367
          ×
3337202759497815655622601060535511422794076034476755466678452098702384172921003708025744867329688187756571898625803693206 2711
```

RSA factoring challenge
(Product of exactly two primes)

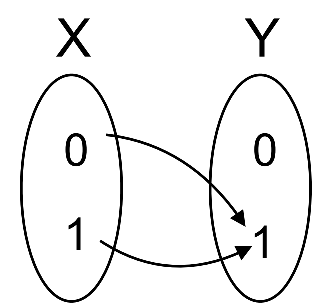


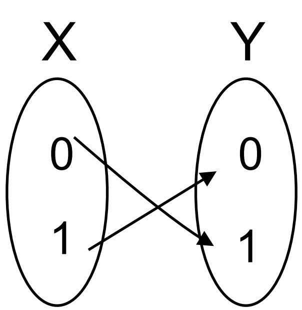
Deutsch Algorithm

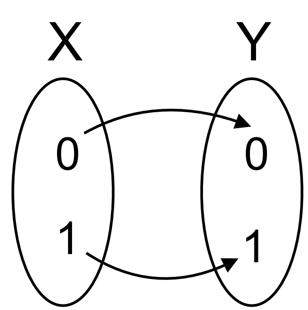
1985

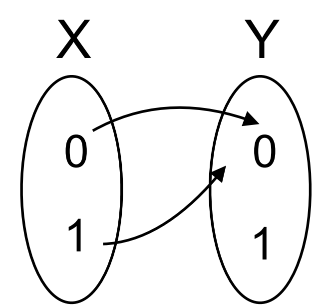
- We want to find out whether a particular function, with one input bit and one output bit is constant or balanced. Classically, we need to evaluate the function twice (i.e., for input = 0 and input = 1), but remarkably, we only need to evaluate the function once using quantum algorithm, by using Deutsch's algorithm.

- Consider a simple function, $f(x) : \{0,1\} \longrightarrow \{0,1\}$
- For possible functions
  - Identity: $f(0) = 0$ and $f(1) = 1$
  - Bit-flip function: $f(0) = 1$ and $f(1) = 0$
  - Constant function: $f(x) = 0$ or $f(x) = 1$





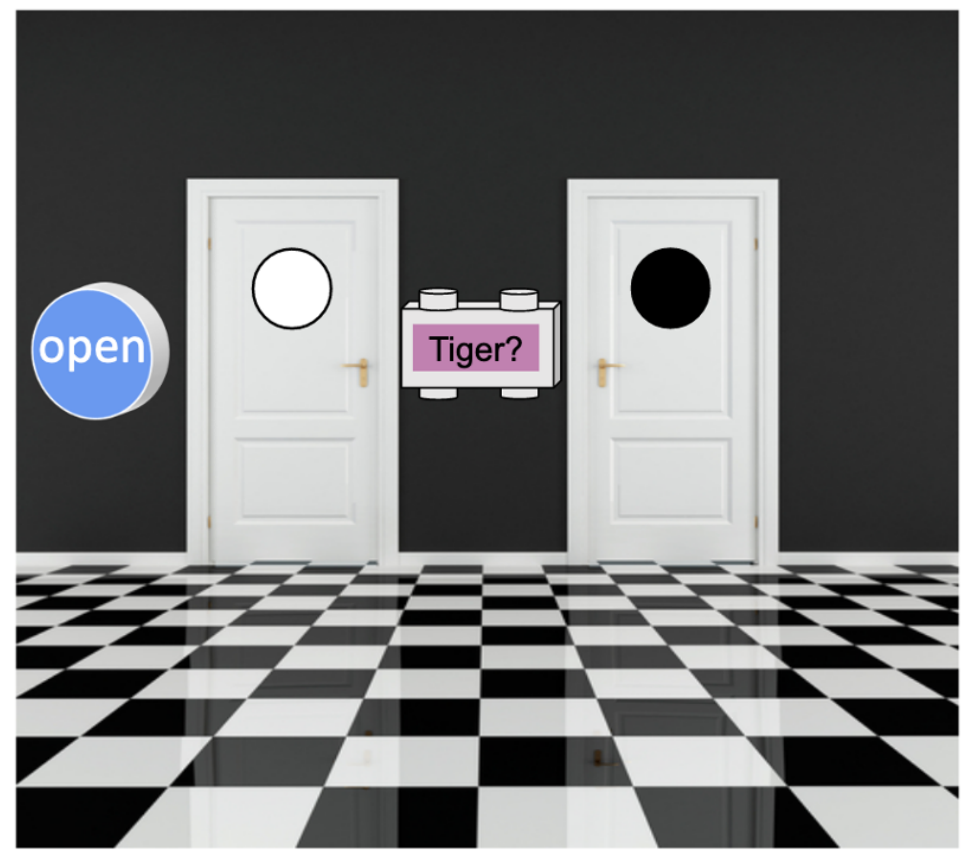




$ vs Tiger

- Teaching quantum information science to high-school and early undergraduate students by Sophia Economou, Terry Rudolph, Edwin Barnes, 2005.07874

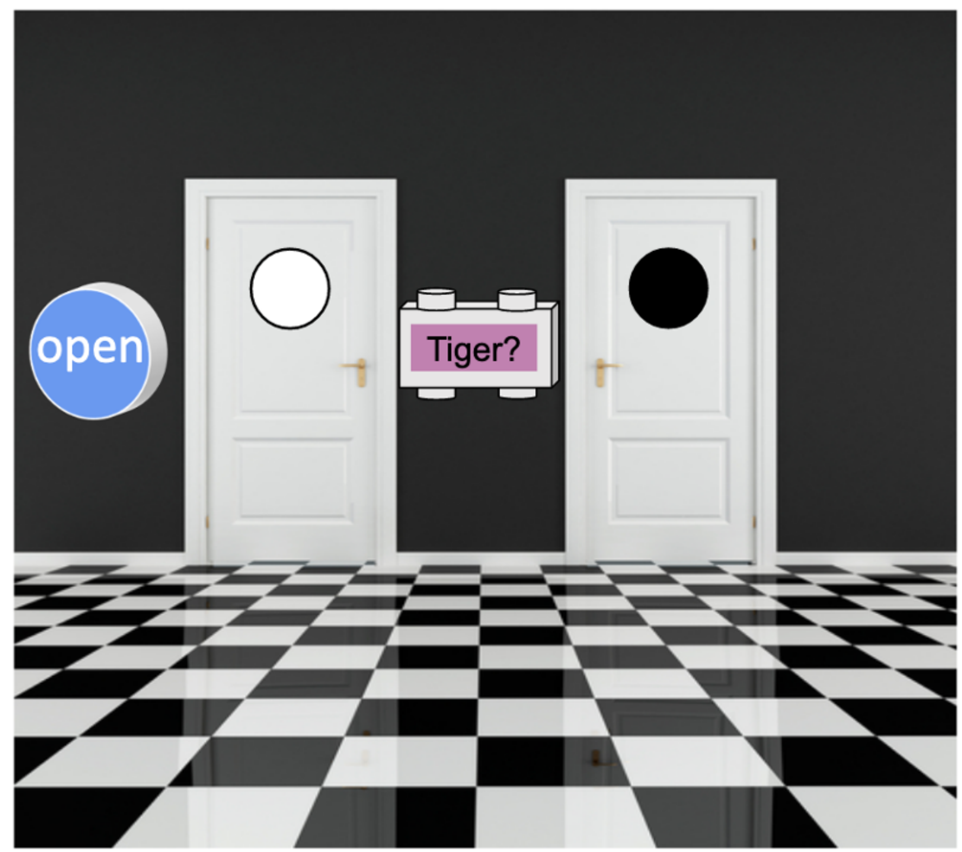
- You encounter two doors: Money behind at least one door



- Tiger might be lurking behind one door

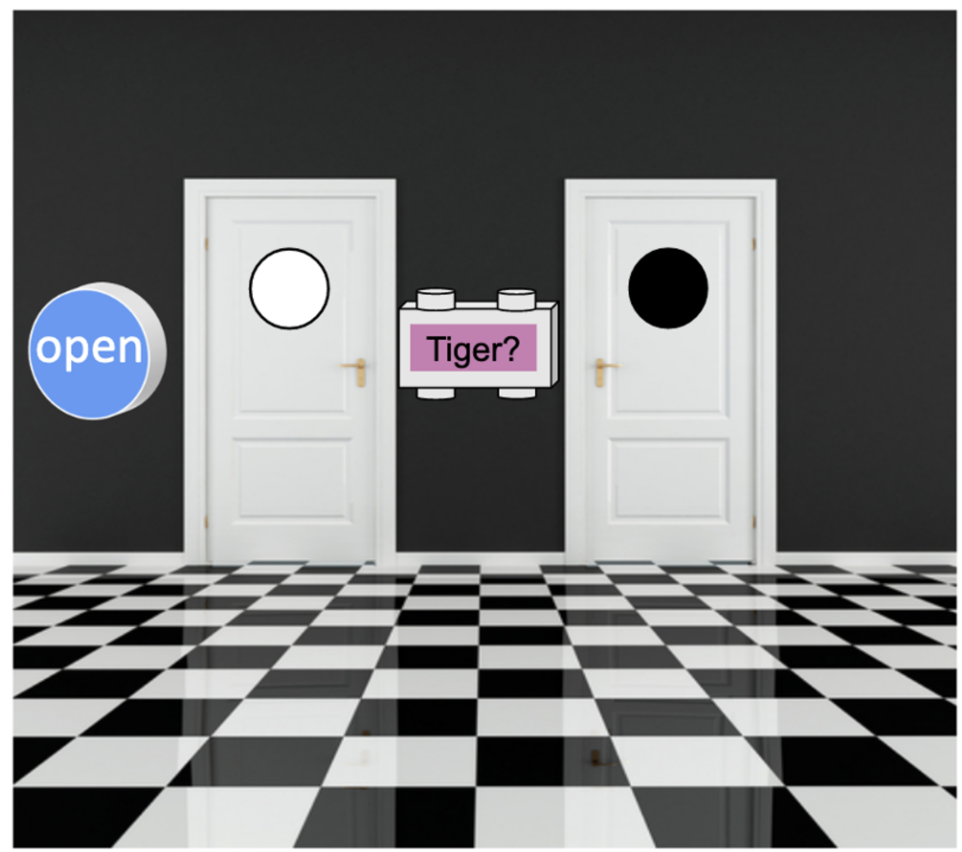
$ vs Tiger

- The button on the left opens both doors
- YOU WANT TO BE SURE THERE'S NO TIGER BEHIND EITHER DOOR BEFORE YOU PUSH THE "OPEN" BUTTON
- The device in the middle will tell you if there is a tiger behind the door that you ask about – but you only get to use it once



$ vs Tiger

- List the three different scenarios for what’s behind the doors:



$ vs Tiger

- List the three different scenarios for what's behind the doors:

$ vs Tiger

- Make a truth table for the tiger box for each of the scenarios

	In	Out
Door Open?	0 1	
Door Open?	1 1	

In	Out
0 1	
1 1	

In	Out
0 1	
1 1	

$ vs Tiger

- Make a truth table for the tiger box for each of the scenarios

	In	Out	In	Out	In	Out
Door Open?	0 1	0 1	0 1	0 1	0 1	0 0
Door Open?	1 1	1 0	1 1	1 1	1 1	1 1

1 = no tiger
0 = tiger

$ vs Tiger

- What gate(s) correspond to the truth table for each scenario?

	In	Out
Door	0	0
Open?	1	1
Door	1	1
Open?	1	0

In	Out
0	0
1	1
1	1
1	1

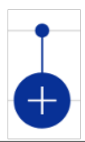
In	Out
0	0
1	0
1	1
1	1

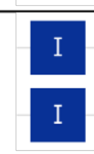
1 = no tiger
0 = tiger

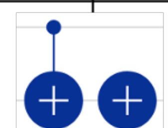
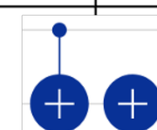
$ vs Tiger

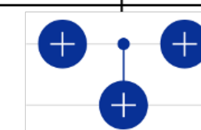
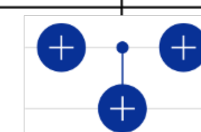
- What gate(s) correspond to the truth table for each scenario?

1 = no tiger
0 = tiger

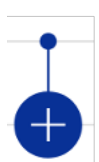
	In		Out
Door	0		0
Open?	1		1
Door	1		1
Open?	1		0

In		Out
0		0
1		1
1		1
1		1

In		Out
0		0
1		0
1		1
1		1

In		Out
0		0
1		0
1		1
1		1

Conditional NOT



Identity

Both are possible.

$ vs Tiger

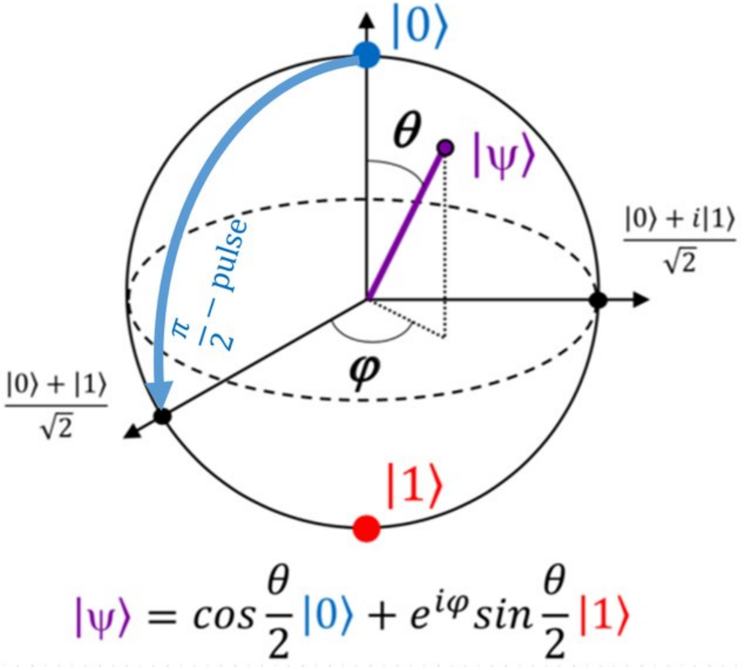
- Challenge: We can't change the tiger box, but can we add gates before and/or after it such that we can determine if there is a tiger somewhere by ONLY USING THE TIGER BOX ONCE?

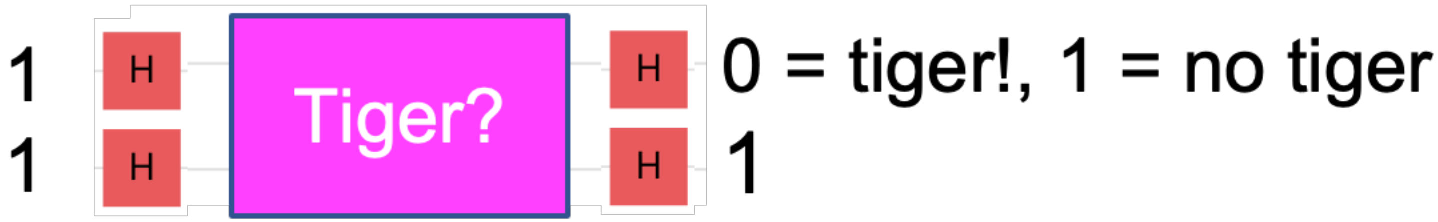
- We're trying to prove that quantum computing let's us do things that are impossible with classical computing. Therefore, consider adding some quantum gates.

- Hint 1: We'd like to query both doors with one push of a button, so maybe we should put the "Door" bit into a superposition.
- Hint 2: We definitely don't want a superposition output, so we maybe we should add a second H to the "Door" bit.
- Hint 3: Our inputs will always be 11 for the solution.
- Hint 4: We want the output to be 11 for no tiger and 10 for tiger.

$ vs Tiger

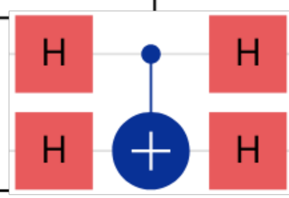
- H changes 0 into + state.
- H changes 1 into - state.



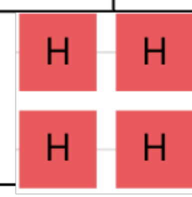


$ vs Tiger

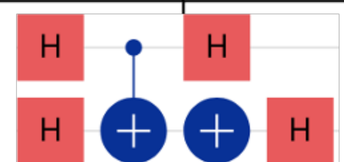
In	Out
1	0
1	1



In	Out
1	1
1	1



In	Out
1	0
1	1

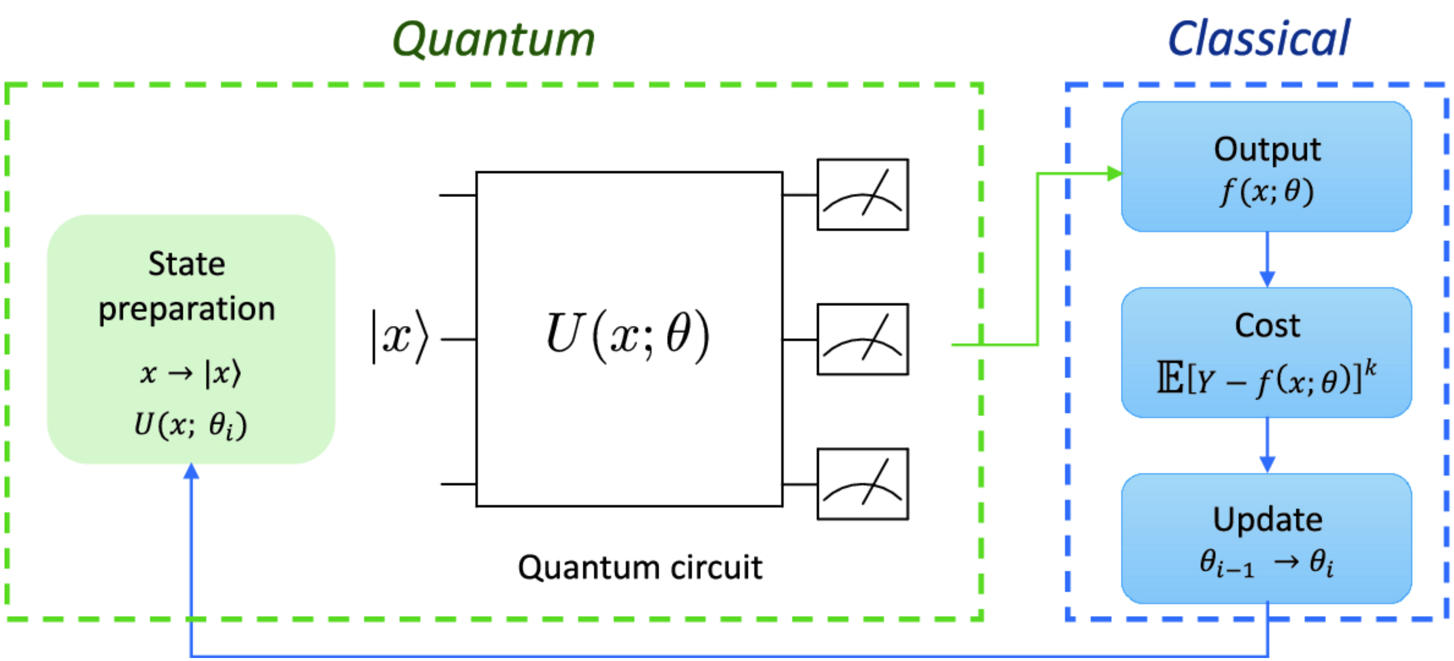


- What does this tell us?
  - We can solve (some) unsolvable problems with quantum computing
  - We can determine IF there is a tiger, but not WHICH DOOR

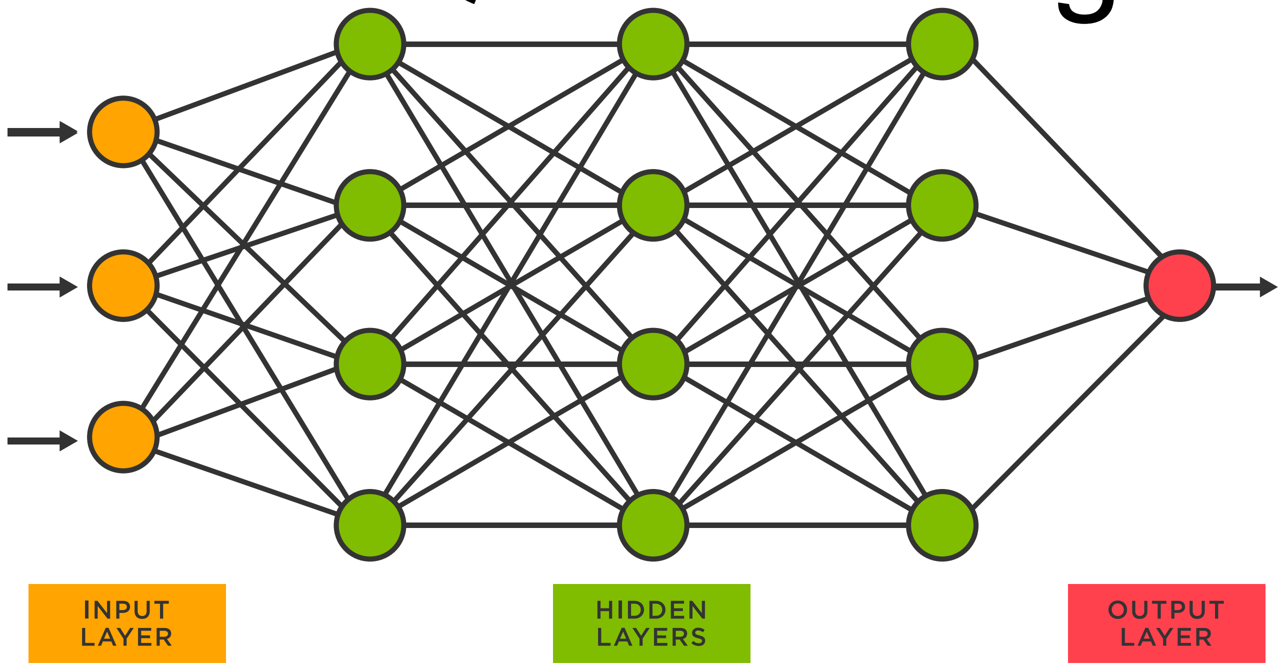
Variational Quantum Algorithms

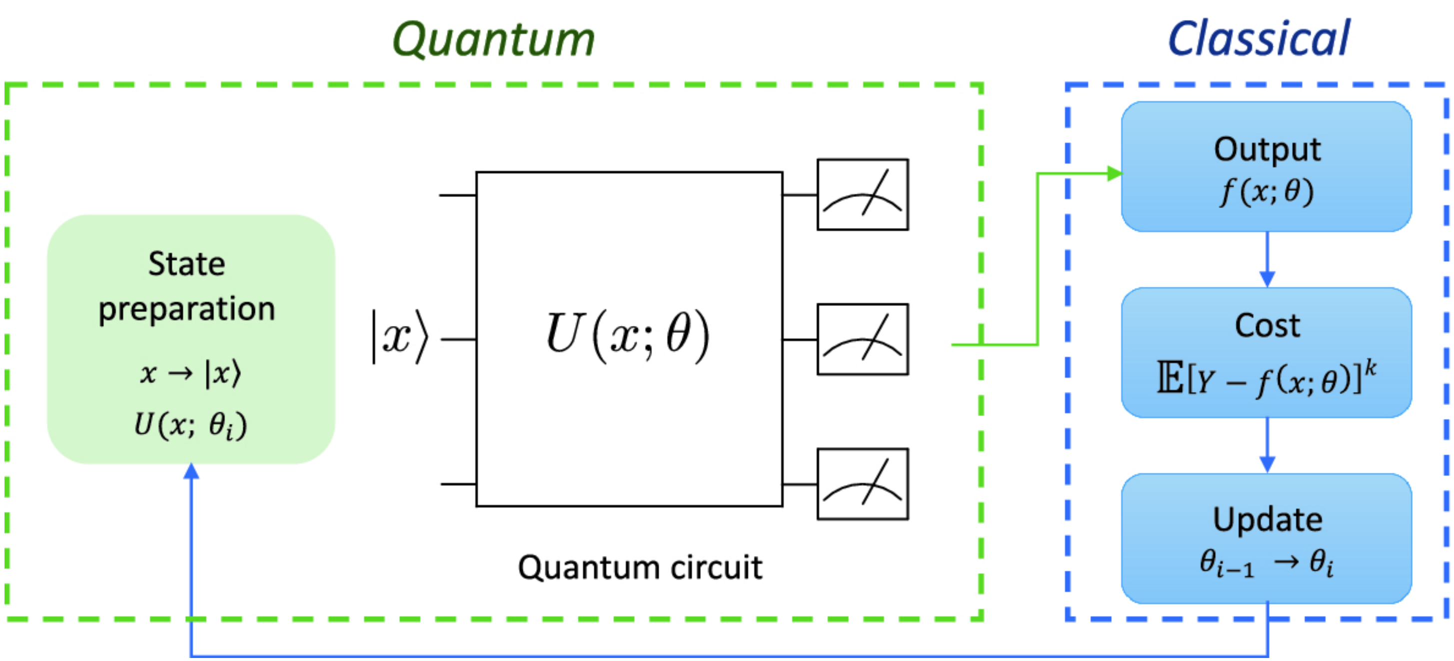
- Hybrid quantum-classical model is suggested to circumvent the issue of going slow with quantum annealer as well as implementing Hamiltonian in the available hardware.
- Quantum: parameterize wave function
- Classical: minimize/maximize the expectation value of H in the problem.

$$E(\vec{\theta}) = \langle\, \psi(\vec{\theta})\,|\, H \,|\, \psi(\vec{\theta})\,\rangle$$



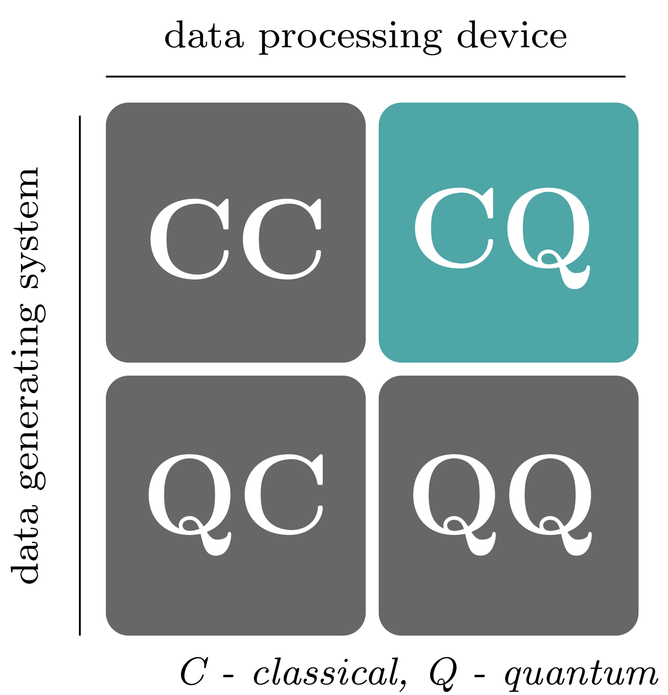
Variational Quantum Algorithms





Quantum Machine Learning

- Artificial Intelligence: Statistical prediction
- Machine Learning: Learn from data
- Quantum Machine Learning: Learn from data with quantum algorithms
  - Subdiscipline of quantum computing and quantum information science



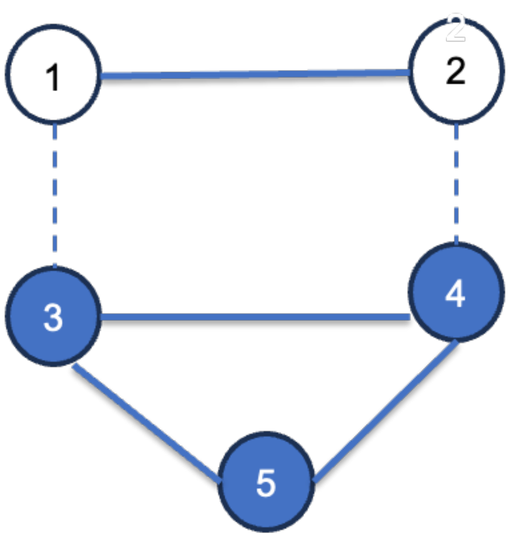
- CC: classical data being processed classically
- QC: how machine learning can help with quantum computing
- CQ: classical data fed into quantum computer for analysis (quantum machine learning)
- QQ: quantum data being processed by quantum computer (ex: Quantum simulation)

Quantum Optimization

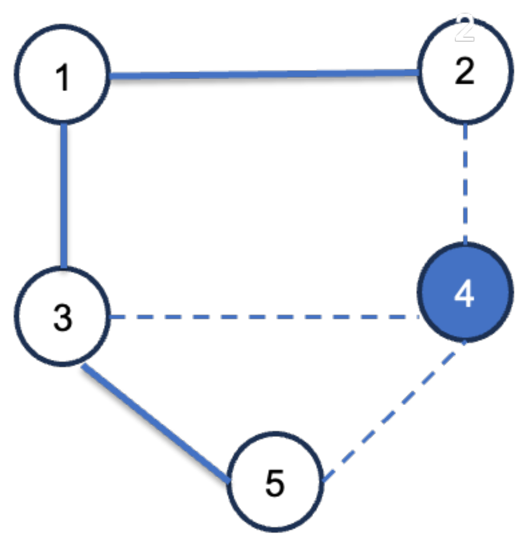
- Optimization problems are everywhere.
  - Continuous optimization
  - Discrete optimization: combinatorial optimization
  - Quadratic Unconstrained Binary Optimization (QUBO)
    - NP hard problem
    - Quadratic function might have several local minima
    - Close connection to Ising model
- Apply quantum algorithms to solve optimization problem
  - Gate model: use universal gates (Pauli’s)
  - Non-gate model (quantum annealer): relies on adiabatic theorem to find a minimum energy of Hamiltonian corresponding to the minimum value of some cost function.

MaxCut problem

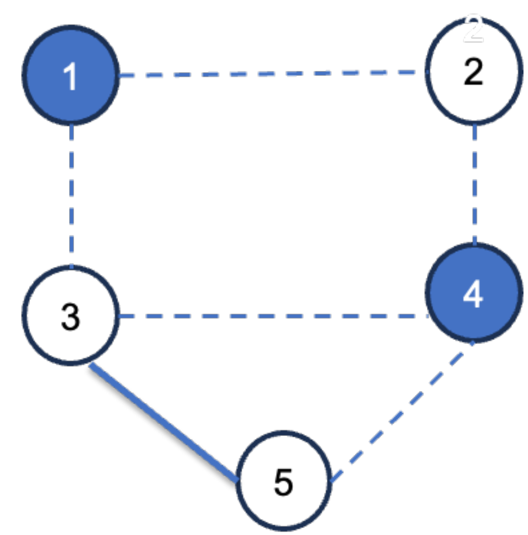
- Objective:
  - Maximize the number of cut edges in a graph when split into 2 parts
  - Divides the set of nodes in the graph into two subsets so that we have as many edges as possible that go between the two sets



Cut size = 2



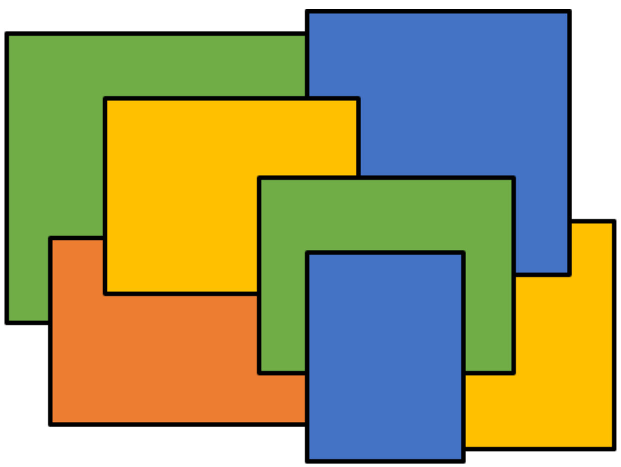
Cut size = 3



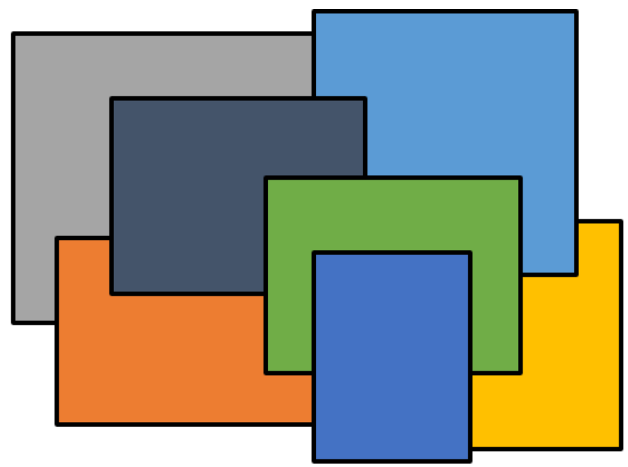
Cut size = 5

Map Coloring

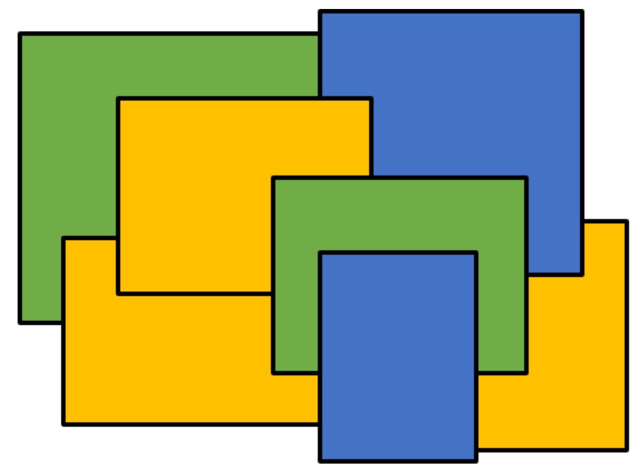
- Problem: Color the regions of a given map such that
  - at most four colors are used
  - no two adjacent* regions have the same color



Valid four-coloring



Invalid; more than 4 colors

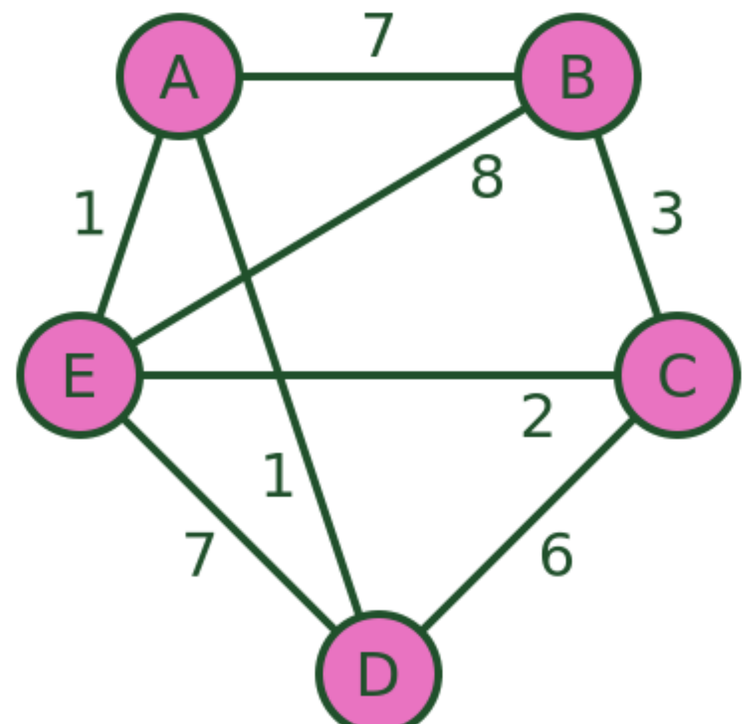


Invalid; yellows touch

* "adjacent" means "shares an edge"; sharing a corner is OK

Traveling salesman problem (TSP)

- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- It is an NP-hard problem in combinatorial optimization, important in theoretical computer science and operations research.



Quadratic Unconstrained Binary Optimization (QUBO)

- QUBO: combinatorial optimization problem with a wide range of applications from finance to ML (partitioning, graph coloring, task allocation, max-sat, max-cut etc)

$$f : \mathbb{Z}_2^n \longrightarrow \mathbb{R}$$

Quadratic polynomial over binary variable

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{i} q_{ij}\, x_i\, x_j + \sum_{i=1}^{n} h_i\, x_i$$

$$x_i \in \mathbb{Z}_2 = \{0,1\}, \quad h_i, q_{ij} \in \mathbb{R}$$

$$x = (x_n x_{n-1} \cdots x_2 x_1)$$

- Find a binary vector x^* which minimizes f (binary strings of n-bits)

$$x^* = \underset{x \in \mathbb{Z}_2^n}{\operatorname{argmin}}\, f(x)$$

- In matrix notation, $f(x) = x^T Q x,$ where $Q \in \mathbb{R}^{n \times n}$
- NP-hard problem: no polynomial-time algorithms are known.

Ising formulations of many NP problems

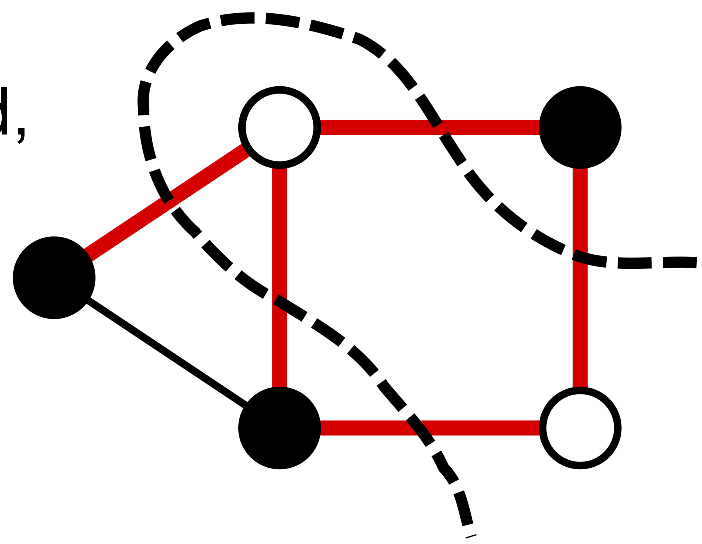
Andrew Lucas

1302.5843

$$H = -\sum_{i,j} J_{ij}\,\sigma_i^z\,\sigma_j^z - \sum_i h_i\,\sigma_i^z$$

QUBO example: Max-cut Problem

- Max-Cut is the NP-hard problem of finding a partition of the graph's vertices into an two distinct sets that maximizes the number of edges between the two sets.
- Undirected Graph: G = (V, E)
  - V: set of nodes, and E: set of edges
- Partition vertices into two complementary sets such that the number of edges between the two sets is as large as possible.
- As the Max-Cut Problem is NP-hard, no polynomial-time algorithms for Max-Cut in general graphs are known.



QUBO example: Max-cut Problem

- The cost function to be maximized:

$$C(x) = \sum_{(i,j)\in E} \left(x_i + x_j - 2x_i x_j\right) \quad \text{where } x_i \in \{0,1\}$$

$$x_i + x_j - 2x_i x_j = 1, \text{ if } x_i \text{ and } x_j \text{ belong in different sets}.$$

$s_i \in Z_2 = \{-1,1\}$

$$x_i + x_j - 2x_i x_j = 0, \text{ if } x_i \text{ and } x_j \text{ belong in the same set}.$$

- Introducing $x_i = \dfrac{s_i + 1}{2}$, the cost function can be rewritten

$$C(s) = \frac{1}{2}\sum_{(i,j)\in E}\left(1 - s_i s_j\right) \longrightarrow C(s) = \frac{1}{2}\sum_{(i,j)\in E}\left(1 - \sigma_i^z \sigma_j^z\right)$$

(i,j) : the edge index

i : vertex index

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \sigma^z|0\rangle = +1|0\rangle \quad \sigma^z|1\rangle = -1|1\rangle$$

σ_i^z : Pauli's Z matrix actingon the i^{th} vertex

σ_j^z : Pauli's Z matrix actingon the j^{th} vertex

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

Matrices = linear operators = observables

Eigenvalues = what are actually measured in experiments

Combinatorial problems at the LHC

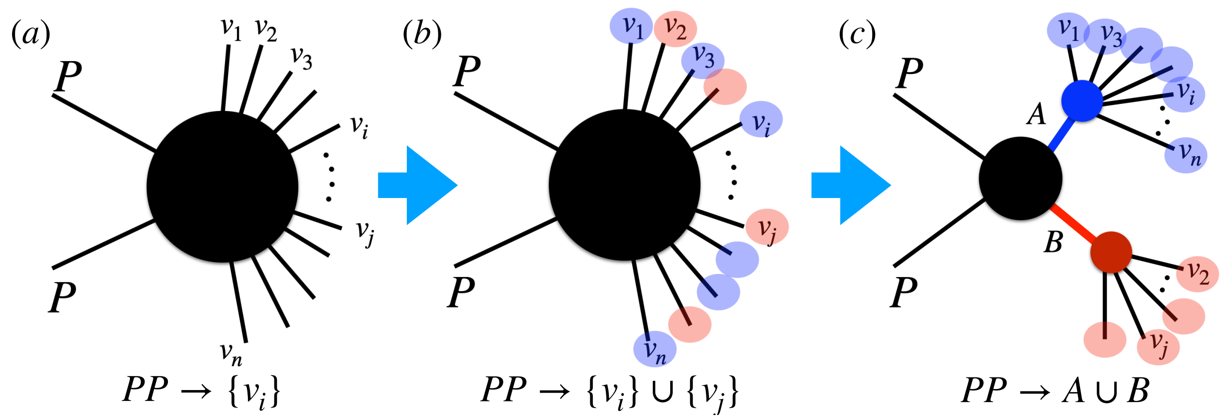


FIG. 1. (a) n-observed particles (b) Dividing n particles into two groups for $2 \to 2$ process (c) Identified event-topology with A and B.

- Assuming $2 \to 2$ production with subsequent decays, identification of an event-topology becomes a binary classification, with 2^{n-1} possibilities.
- Combinatorial problem: What would be an efficient way of assigning all observed particles in two decay chains?

p_i is the momentum of constituent of A if $x_i = 1$ $\qquad P_1 = \sum_i p_i\, x_i$

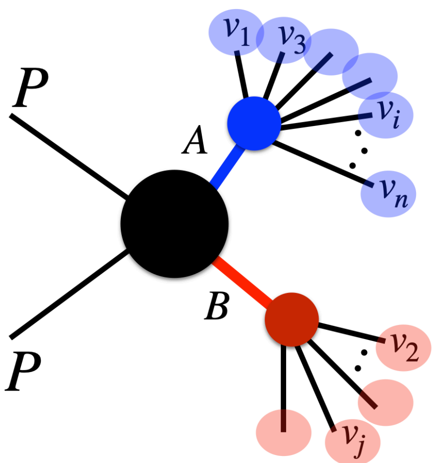
p_i is the momentum of constituent of B if $x_i = 0$ $\qquad P_2 = \sum_i p_i\,(1 - x_i)$

Minimize the mass difference: $H = \left(P_1^2 - P_2^2\right)^2$ for all possible combinations of x_i

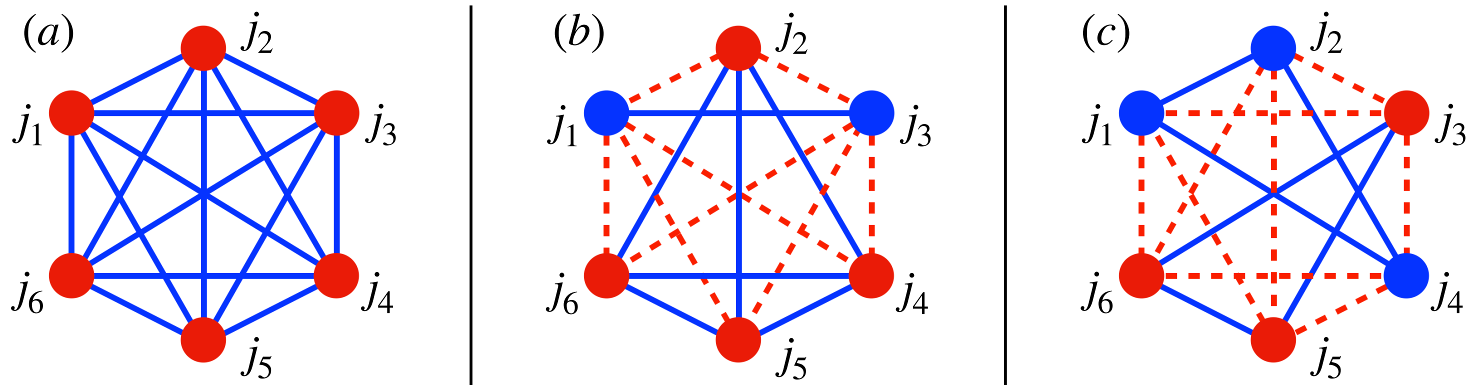
Hybrid quantum-classical approach for combinatorial problems at hadron colliders, 2410.22417

Leveraging Quantum Annealer to identify an Event-topology at High Energy Colliders, 2111.07806

Combinatorial problems in the top quark production



$$PP \to A \cup B$$



$$H_0 = \left(P_1^2 - P_2^2\right)^2$$

$$P_1 = \sum_{i=1}^{n} p_i\, x_i\,,$$

$$P_2 = \sum_{i=1}^{n} p_i\,(1-x_i)\,,$$

$$H_P = H_0 + \lambda H_1$$

$$\begin{aligned} H_0 &= \left(\sum_{ij} P_{ij} x_i x_j - \sum_{ij} P_{ij}(1-x_i)(1-x_j)\right)^2 \\ &= \left(\frac{1}{4}\sum_{ij} P_{ij}\left[(1+s_i)(1+s_j) - (1-s_i)(1-s_j)\right]\right)^2 \\ &= \left(\sum_{ij} P_{ij} s_i\right)^2 \boxed{= \sum_{ij} J_{ij} s_i s_j\,,} \end{aligned}$$

$$\begin{aligned} H_1 &= (P_1^2 + P_2^2) \\ &= \frac{1}{4}\sum_{ij} P_{ij}\left[(1+s_i)(1+s_j) + (1-s_i)(1-s_j)\right] \\ &\to \frac{1}{2}\sum_{ij} P_{ij} s_i s_j\,, \end{aligned}$$

$$x_i = \frac{1+s_i}{2}$$

$$J_{ij} = \sum_{k\ell} P_{ik} P_{j\ell},$$

$$P_{ij} = p_i \cdot p_j$$

$$\lambda = \frac{\min J_{ij}}{\max P_{ij}}$$

Hybrid quantum-classical approach for combinatorial problems at hadron colliders: 2410.22417

Quantum Approximate Optimization Algorithm (QAOA)

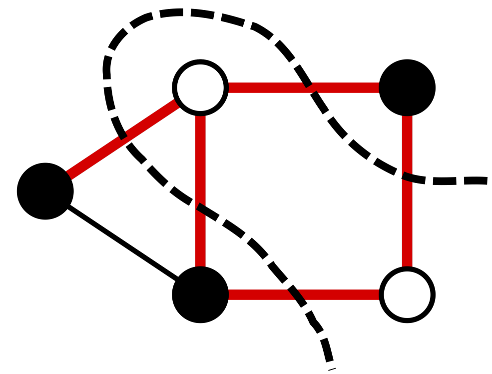
Farhi et al 2014

$$H_P = C(s) = H_C = \frac{1}{2}\sum_{(i,j)\in E}\left(1-\sigma_i^z\sigma_j^z\right) \text{ : Problem Hamiltonian}$$

(i,j) : the edge index

i : vertex index

$$H_M = B = H_B = \sum_j \sigma_j^X \text{ : Mixer Hamiltonian (Initial Hamiltonian)}$$



Undirected Graph: G = (V, E)
V: set of nodes
E: set of edges

Full Hamiltonian: $H(t) = \left(1-\frac{t}{T}\right)H_M + \frac{t}{T}H_P$

$$|\psi\rangle = \exp\left[-i\int_0^t H(t')\,\mathrm{d}t'\right]|\psi_0\rangle = \exp\left[-i\sum_{j=1}^{p} H(j\Delta t)\Delta t\right]|\psi_0\rangle$$

$$\approx \prod_{j=1}^{p}\exp\left[-i\Delta t\left[\left(1-\frac{j\Delta t}{T}\right)H_M + \frac{j\Delta t}{T}H_P\right]\right]|\psi_0\rangle$$

$$\approx \prod_{j=1}^{p}\exp\left[-i\Delta t\left(1-\frac{j\Delta t}{T}\right)H_M\right]\exp\left[-i\Delta t\frac{j\Delta t}{T}H_P\right]|\psi_0\rangle$$

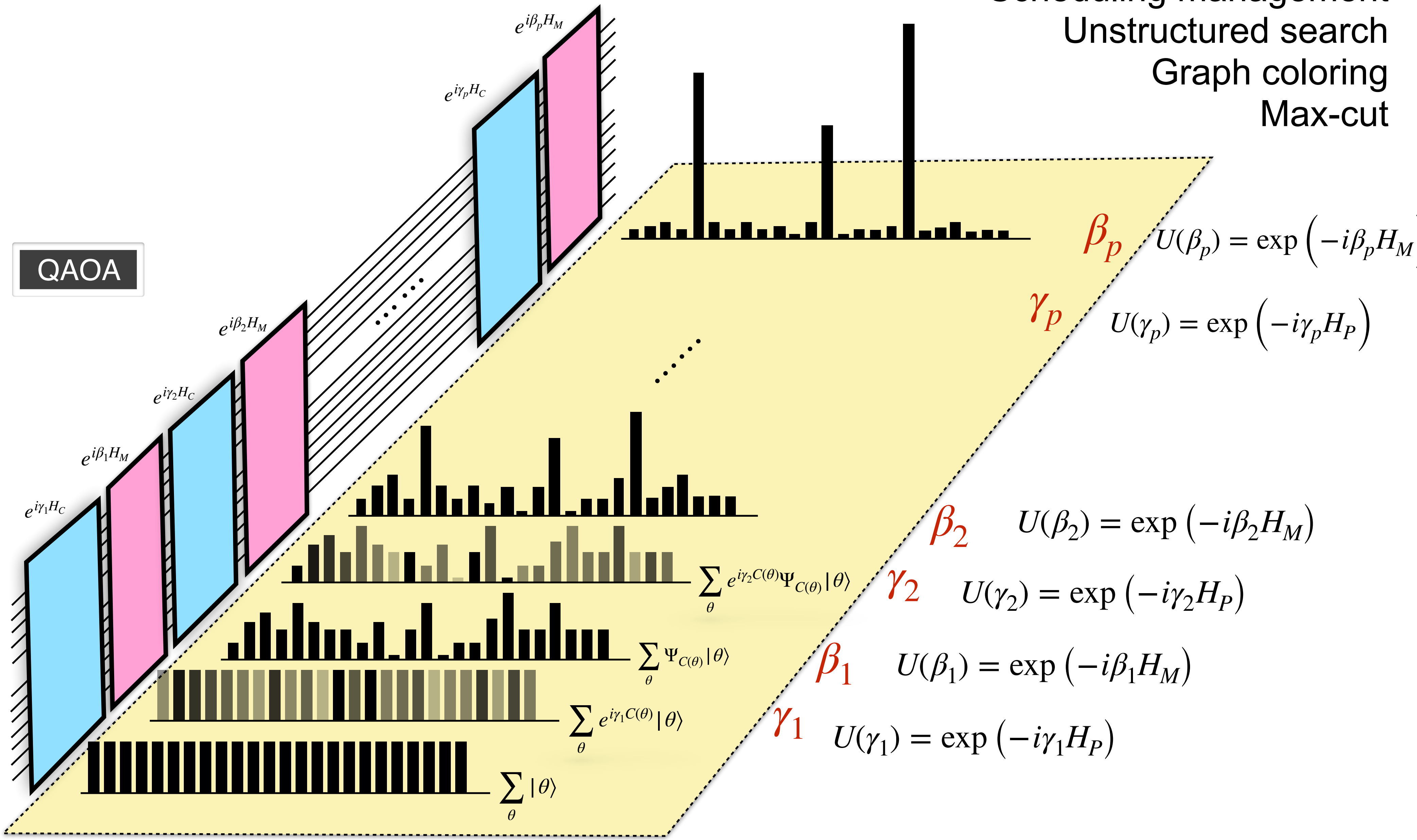
Works in the adiabatic limit or $p \to \infty$

$$= \prod_{j=1}^{p}\underbrace{\exp\left[-i\beta_j H_M\right]}_{U(H_M,\beta_j)}\underbrace{\exp\left[-i\gamma_j H_P\right]}_{U(H_P,\gamma_j)}|\psi_0\rangle$$

$$|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle = \prod_{j=1}^{p} U(H_M,\beta_j)U(H_P,\gamma_j)\,|+\rangle^{\otimes n}$$

QAOA

Maximum Likelihood detection
Traveling salesman problem
Scheduling management
Unstructured search
Graph coloring
Max-cut



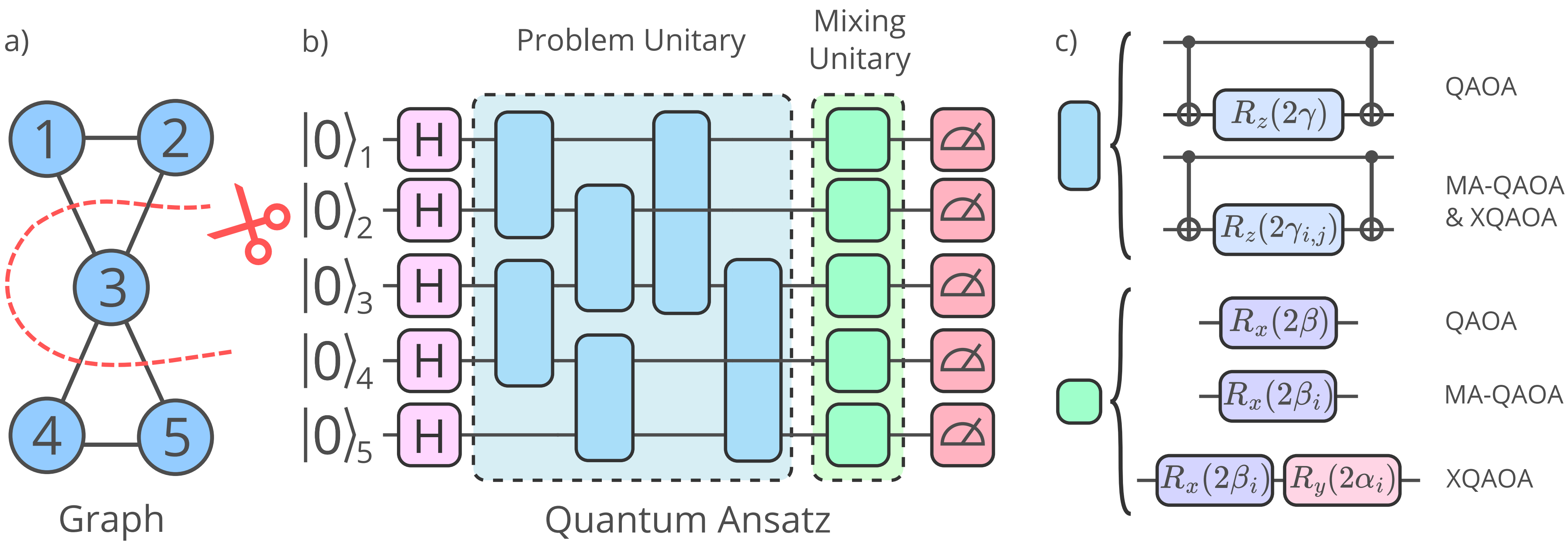


FIG. 1: a) A specific instance of a graph for which we want to identify a set of vertices that maximises the number of edges that are cut. b) A quantum circuit with a single iteration of a quantum ansatz applied to it. The quantum ansatz consists of a unitary operation specific to the problem being solved and a problem-independent mixing unitary. c) Decomposing the problem and mixing unitaries for QAOA, MA-QAOA, and XQAOA into CNOT and single-qubit rotation gates.

- QAOA: one parameter for each mixer layer
- ma-QAOA: n_q parameters for each mixer layer
- XQAOA: $2n_q$ parameters for each mixer layer

Multi-angle Quantum Approximate Optimization Algorithm: 2109.11455

An Expressive Ansatz for Low-Depth Quantum Approximate Optimisation: 2302.04479

Feedback-based ALgorithm Quantum Optimization (FALQON)

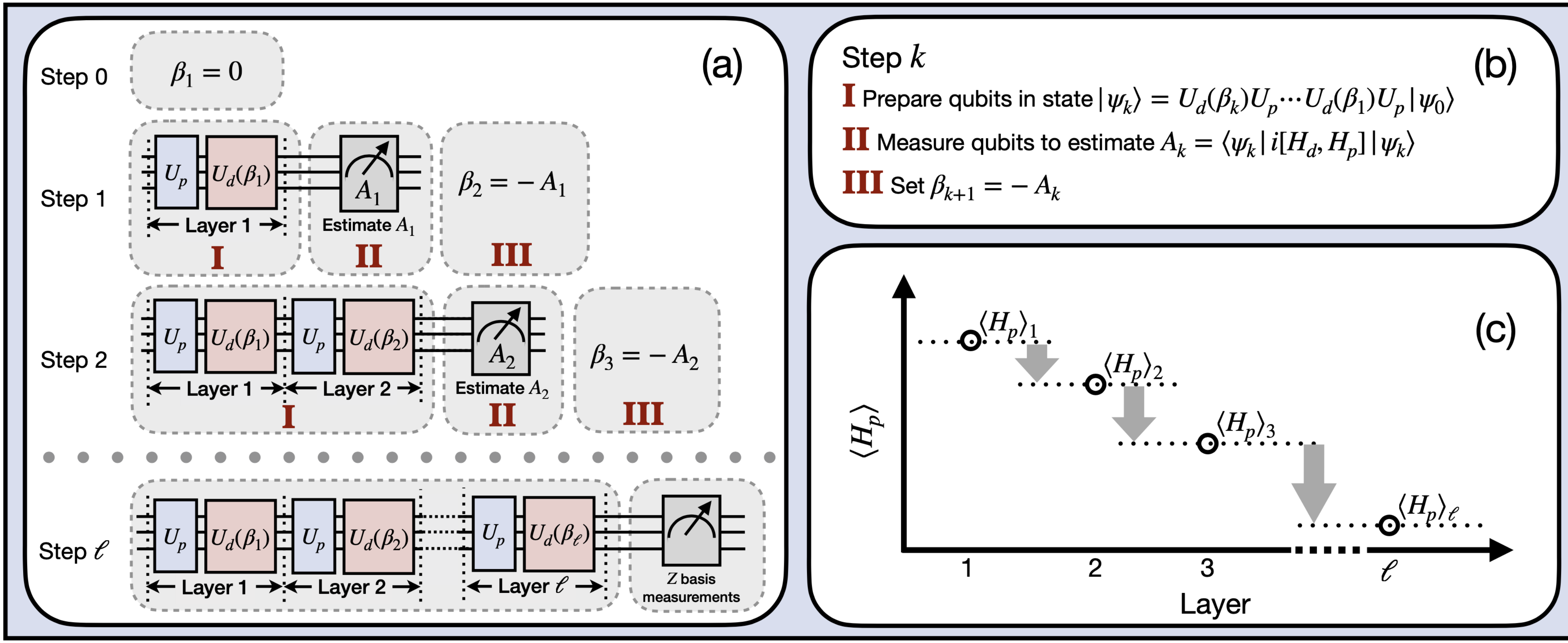
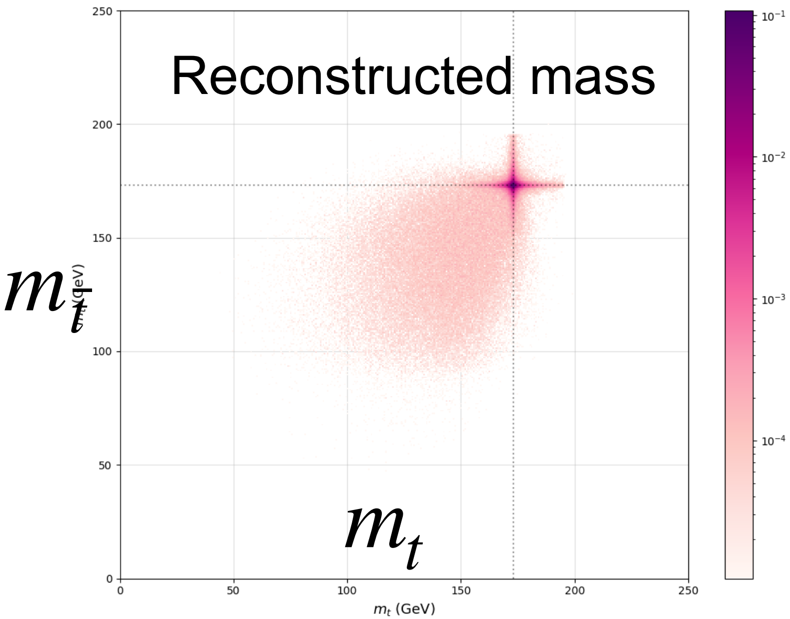
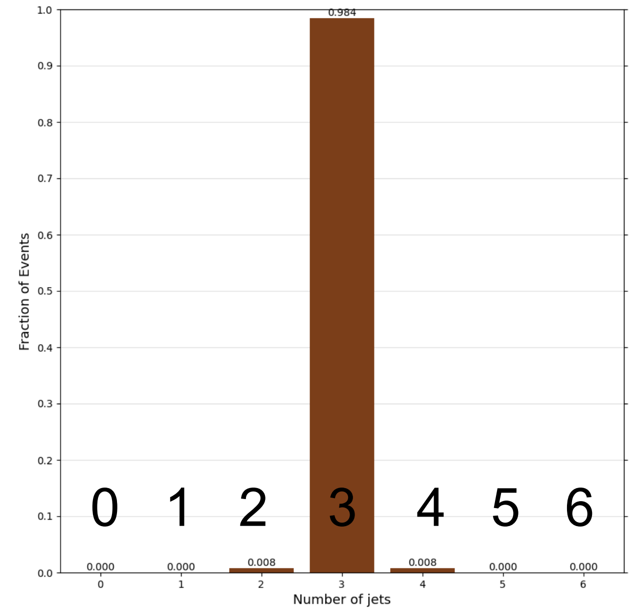


Figure 1. (a) The procedure for implementing FALQON. The initial step is to seed the procedure by setting $\beta_1 = 0$. The qubits are then initialized in the state $|\psi_0\rangle$, and a single FALQON layer is implemented to prepare $|\psi_1\rangle = U_\mathrm{d}(\beta_1)U_\mathrm{p}|\psi_0\rangle$. The qubits are then measured to estimate A_1, whose result is fed back to set $\beta_2 = -A_1$, up to sampling error. For subsequent steps $k = 2, \cdots, \ell$, the same procedure is repeated, as shown in (b): the qubits are initialized as $|\psi_0\rangle$, after which k layers are applied to obtain $|\psi_k\rangle = U_\mathrm{d}(\beta_k)U_\mathrm{p} \cdots U_\mathrm{d}(\beta_1)U_\mathrm{p}|\psi_0\rangle$, and then the qubits are measured to estimate A_k, and the result is fed back to set the value of β_{k+1}. This procedure causes $\langle H_\mathrm{p}\rangle$ to decrease layer-by-layer as per $\langle\psi_1|H_\mathrm{p}|\psi_1\rangle \geq \langle\psi_2|H_\mathrm{p}|\psi_2\rangle \geq \cdots \geq \langle\psi_\ell|H_\mathrm{p}|\psi_\ell\rangle$, as shown in (c), such that the quality of the solution to the combinatorial optimization problem monotonically improves with circuit depth. The protocol can be terminated when the value of $\langle H_\mathrm{p}\rangle$ converges or a threshold number of layers ℓ is reached. Then, after the final step, Z basis measurements on $|\psi_\ell\rangle$ can be used to determine a best candidate solution to the combinatorial optimization problem of interest, by repeatedly sampling from the probability distribution over bit strings induced by $|\psi_\ell\rangle$ and selecting the outcome associated with the best solution.

- Simulated various quantum algorithms using PennyLane to resolve combinatorial ambiguity in the top quark production (6 jets)
  - QAOA, ma-QAOA: Adam optimizer used
  - FALQON: no classical optimization needed
- Compared performance against hemisphere method and classical ML (using SPANet)

Parton-level truth and hemisphere method





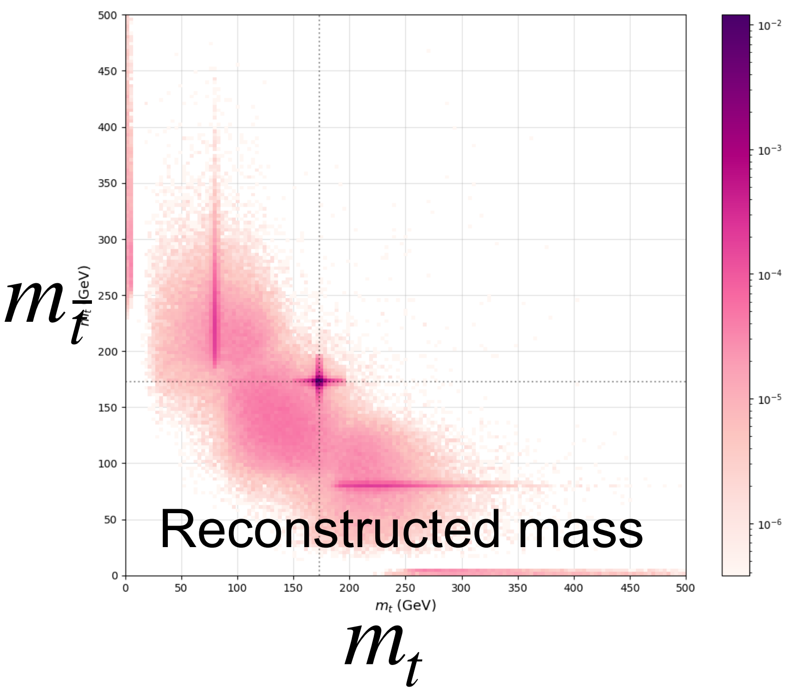
Jet assignment

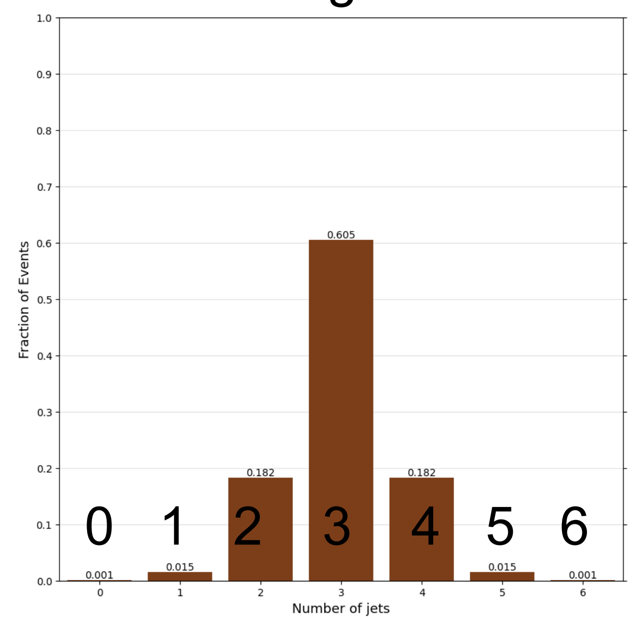
parton-level truth

79% overal efficiency

No mass information used.

Hybrid quantum-classical approach for combinatorial problems at hadron colliders: 2410.22417





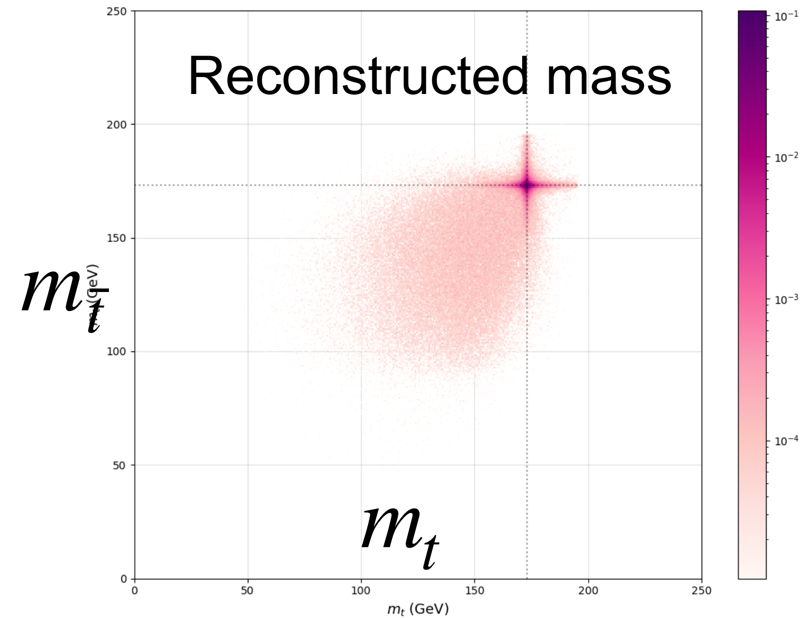
Jet assignment

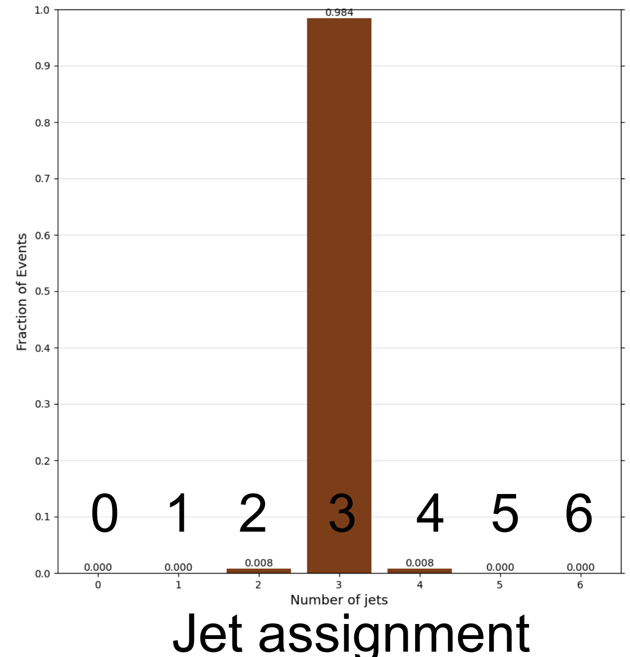
Hemisphere method

50% overal efficiency

No mass information used.

QAOA method



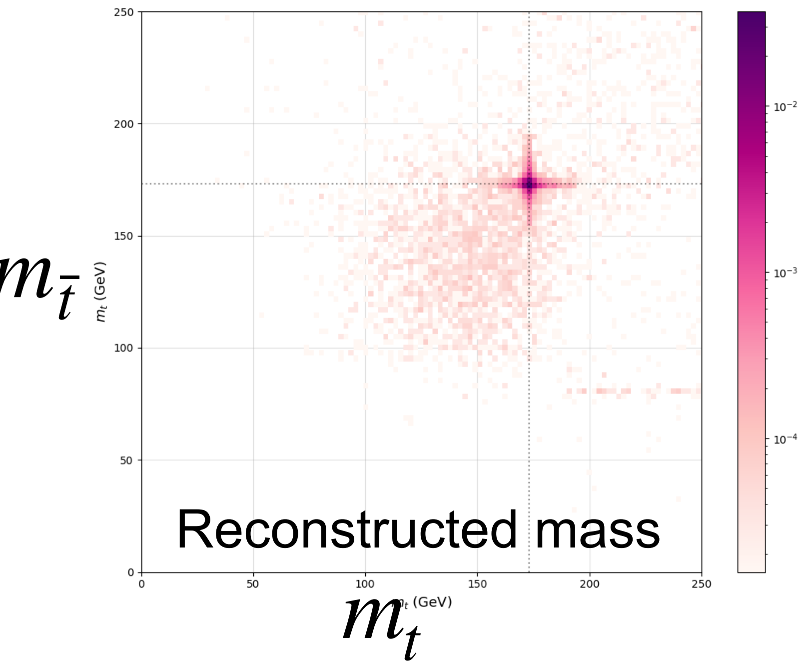


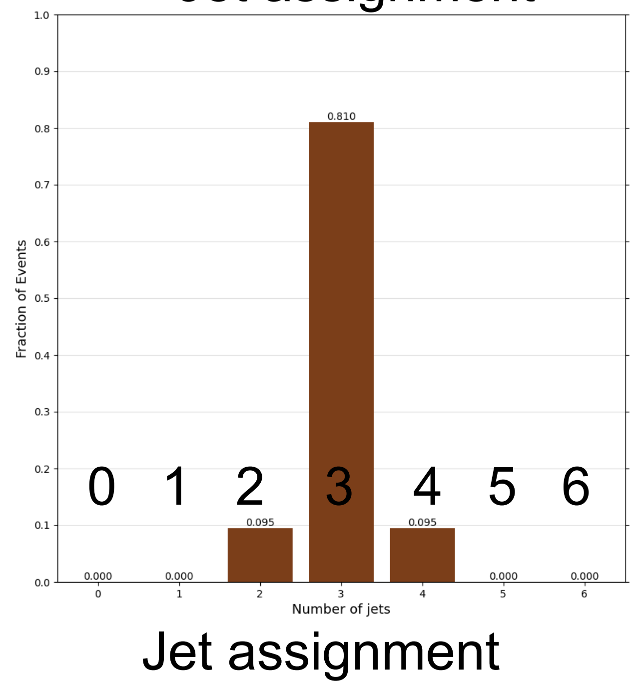
parton-level truth

79% overal efficiency

(Best result expected with the given Hamiltonian)

Hybrid quantum-classical approach for combinatorial problems at hadron colliders: 2410.22417



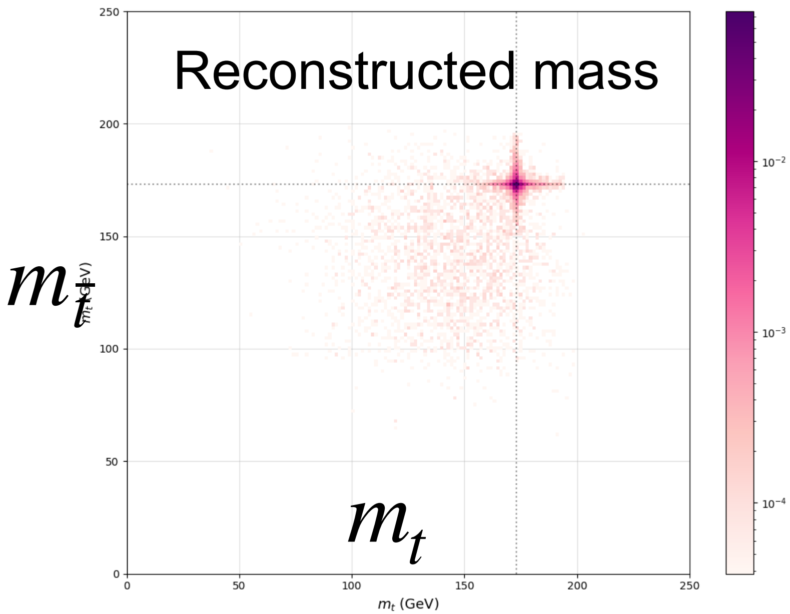


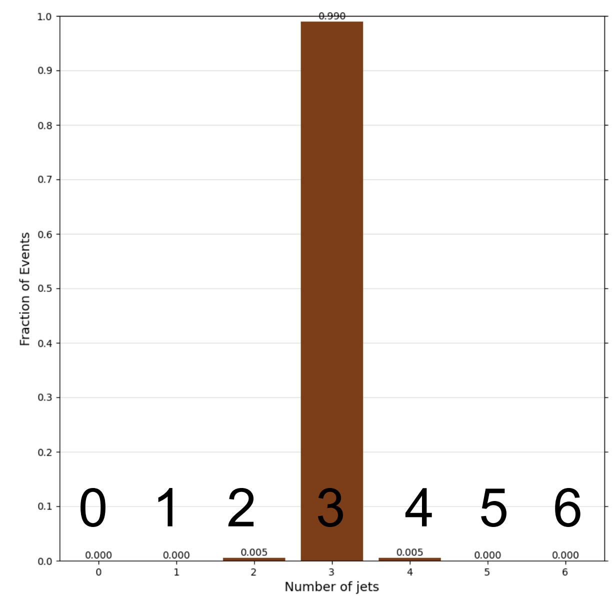
QAOA

55% overal efficiency

No mass information used.

ma-QAOA and FALQON





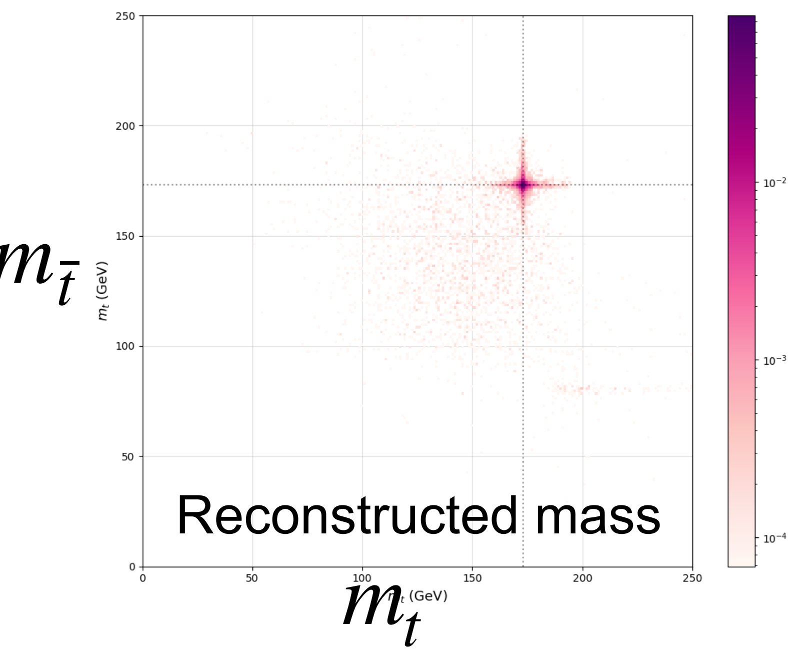
Jet assignment

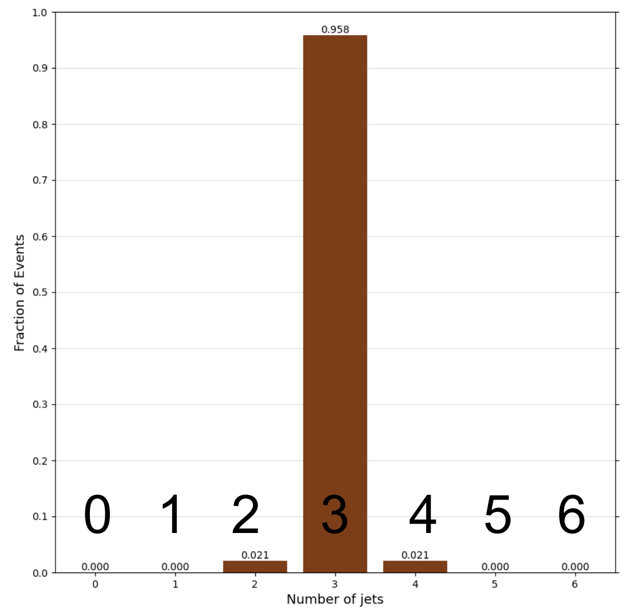
ma-QAOA

75% overal efficiency

No mass information used.

Hybrid quantum-classical approach for combinatorial problems at hadron colliders: 2410.22417





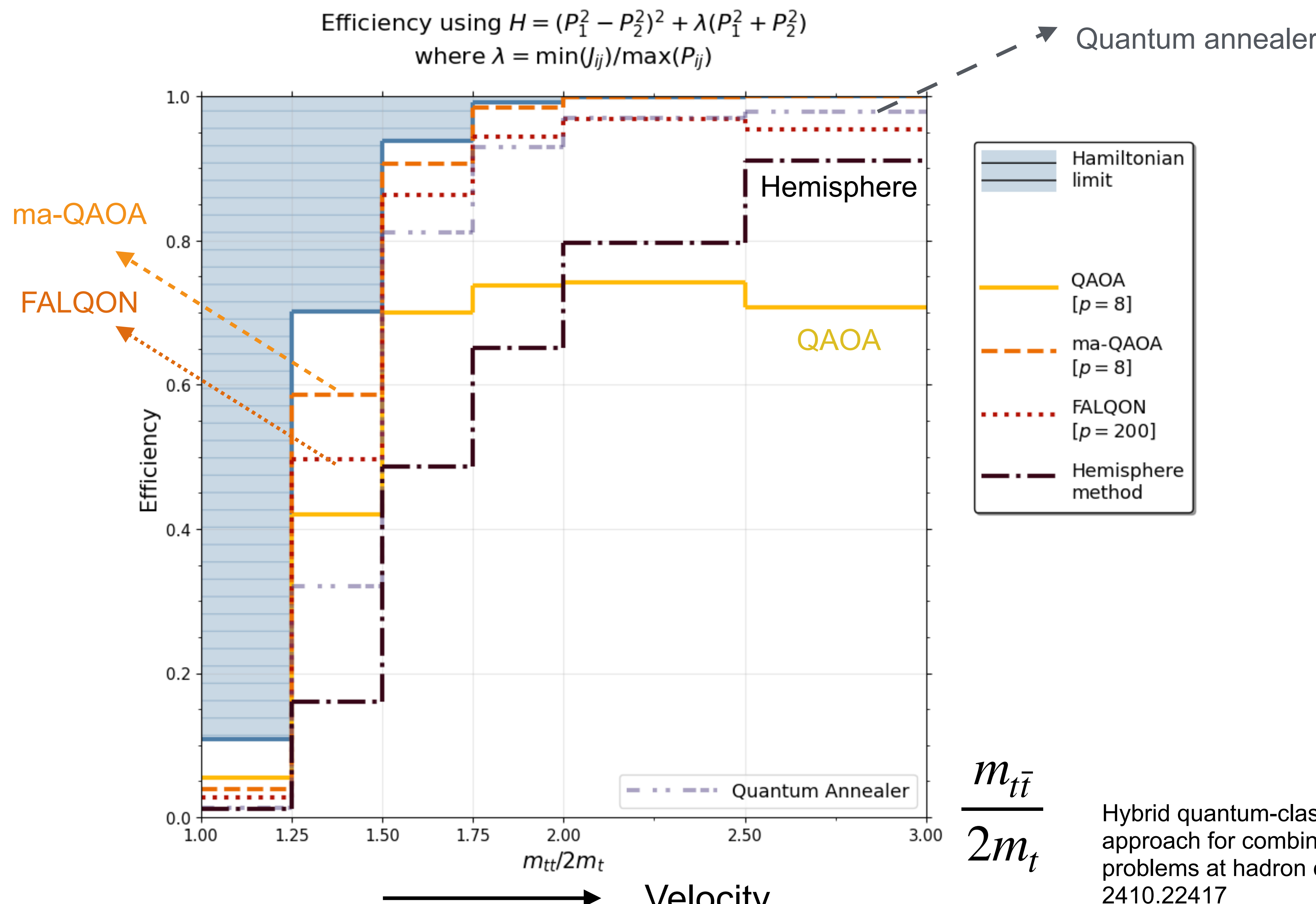
Jet assignment

FALQON

72% overal efficiency

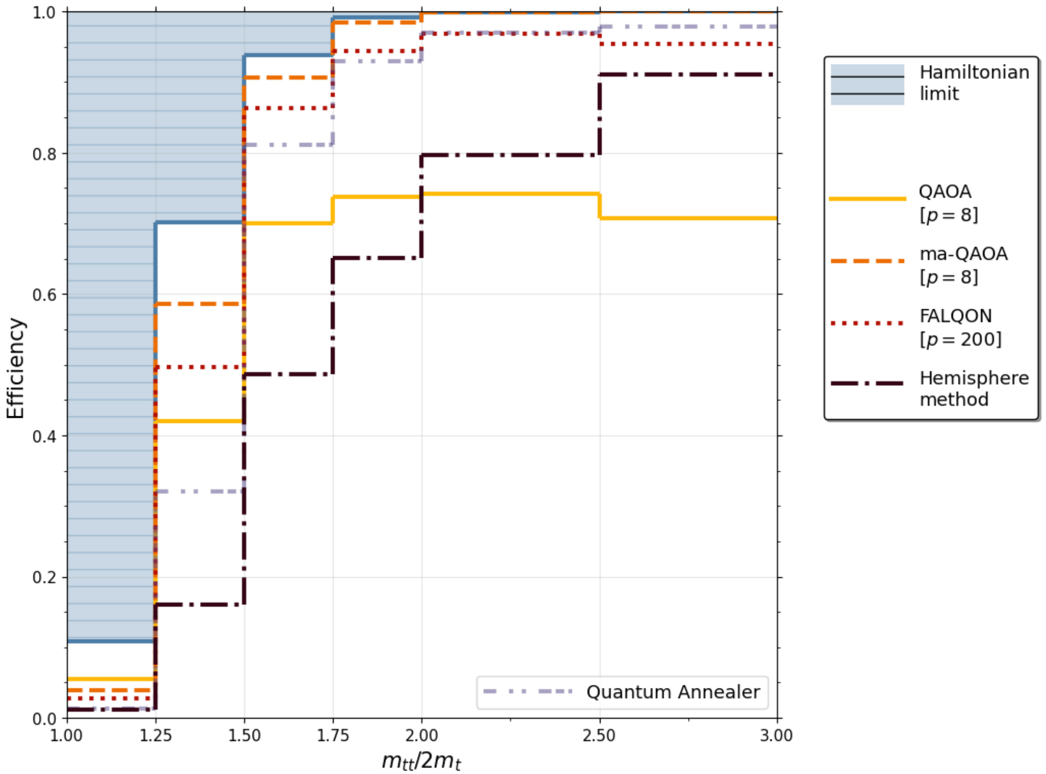
No mass information used.

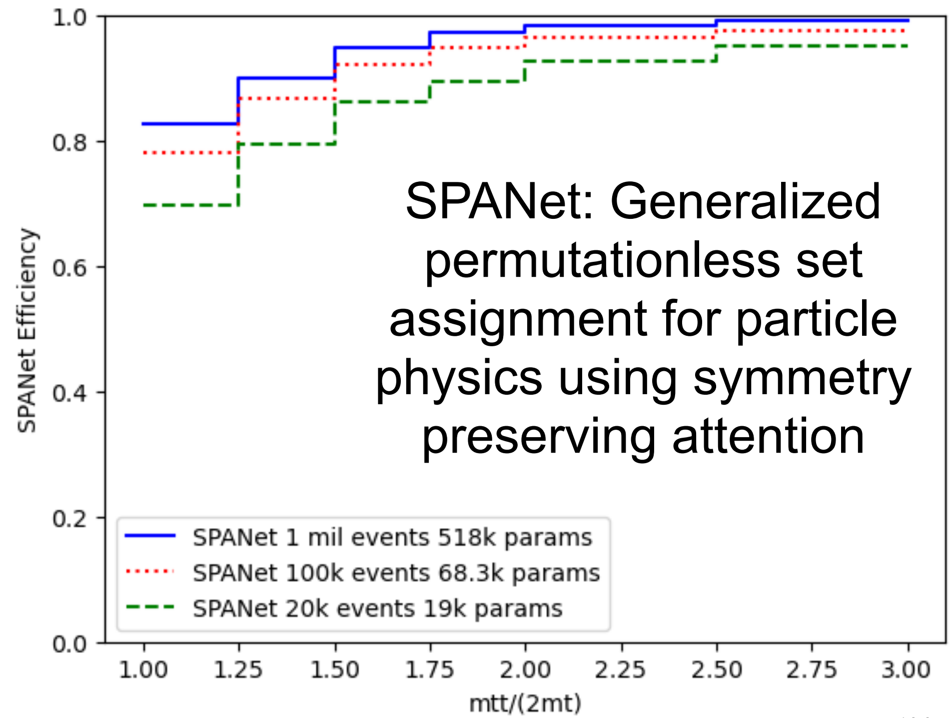
Comparison



Hybrid quantum-classical approach for combinatorial problems at hadron colliders: 2410.22417

Combinatorial problems in the top quark production





$$\frac{m_{t\bar{t}}}{2m_t}$$

Methods	matching accuracy (efficiency)		n_{train}	$n_{\text{parameters}}$	depth (p)	n_{CNOT}	n_{R_Z}	n_{R_X}
	parton-level	smeared events						
Hemisphere	50%	48%	N/A					
QAOA	55%	53%	N/A	16	8	240	120	48
ma-QAOA	75%	73%		168				
FALQON	72%	69%		2	250	7,500	3,750	1,500
VarQITE	79%	??%		15	1	30	15	30
SPANet	91%	70%	5×10^5	10^6	N/A			
	81%	62%	2×10^4	1.9×10^3				

Table 1: Summary of the performance of various methods and the corresponding parameters.

Hybrid quantum-classical approach for combinatorial problems at hadron colliders: 2410.22417

Variational Quantum Imaginary Time Evolution (VarQITE)

- State time evolution under a Hamiltonian: $|\psi(t)\rangle = \sum_m c_m \exp\left(-iE_m \frac{t}{\hbar}\right)|m\rangle$
- Imaginary time, $t \longrightarrow -i\beta$

$$|\psi(\beta)\rangle = \sum_m c_m \exp\left(-E_m \frac{\beta}{\hbar}\right)|m\rangle \underset{\beta \to \infty}{\sim} c_0 \exp\left(-E_0 \frac{\beta}{\hbar}\right)|0\rangle$$

All coefficients decay

$$|0\rangle = \lim_{\beta \to \infty} \frac{|\psi(\beta)\rangle}{\| |\psi(\beta)\rangle \|} \qquad |\psi_{n+1}\rangle = \frac{e^{-\Delta\tau H\hbar}|\psi_n\rangle}{\|e^{-\Delta\tau H\hbar}|\psi_n\rangle\|}$$

- In VarQITE, $|\psi(\beta)\rangle$ is replaced with $|\psi(\theta(\beta))\rangle$ with variational parameters θ. Using McLachlan variational principle, the algorithm update variational parameters θ by minimizing

$$\left|\left|\left(\frac{\partial}{\partial\beta} + H - E_\beta\right)|\psi(\theta(\beta)\rangle\right|\right|$$

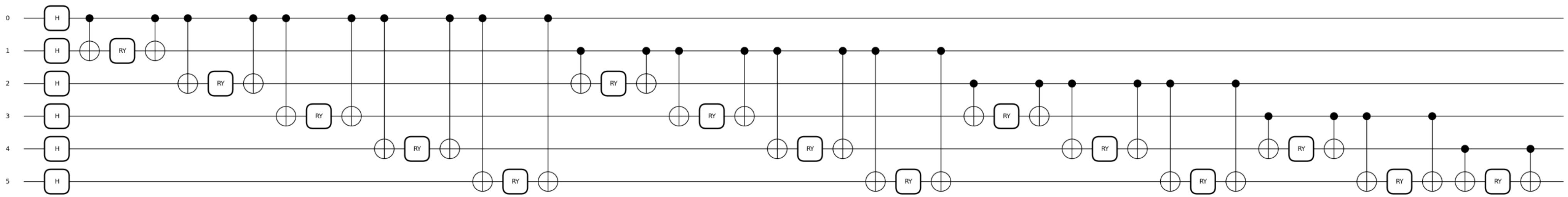
2404.16135, Performant near-term quantum combinatorial optimization

1901.07653, Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution

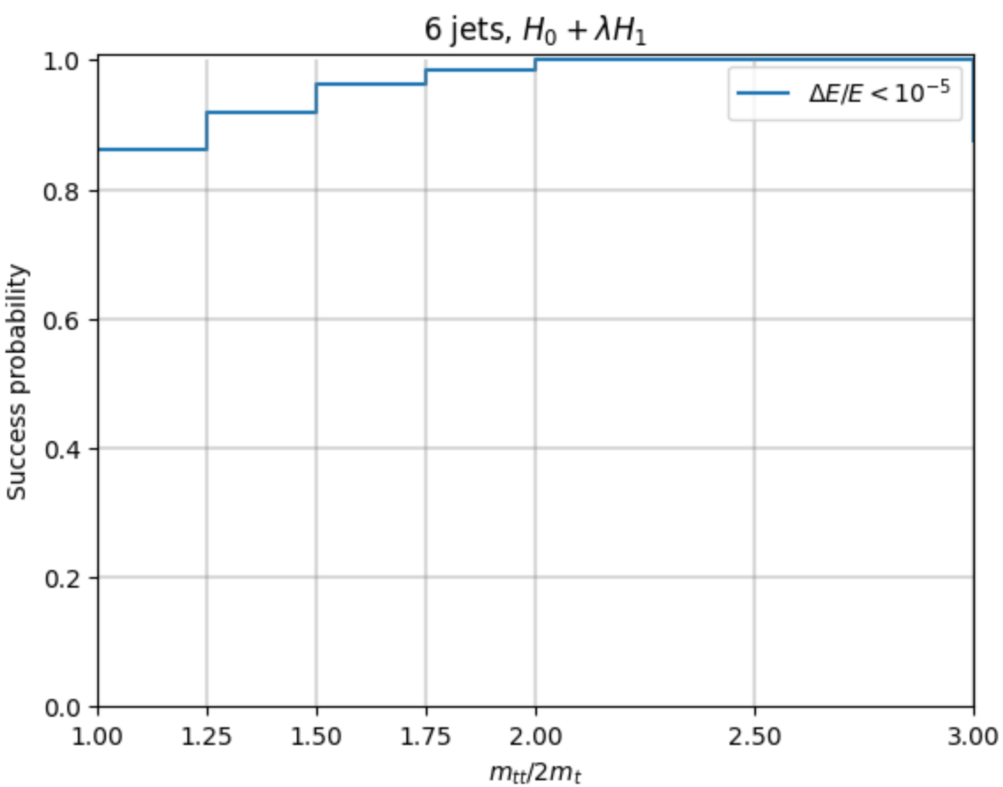
1812.08767, Theory of variational quantum simulation

VarQITE Ansatz (with IonQ)

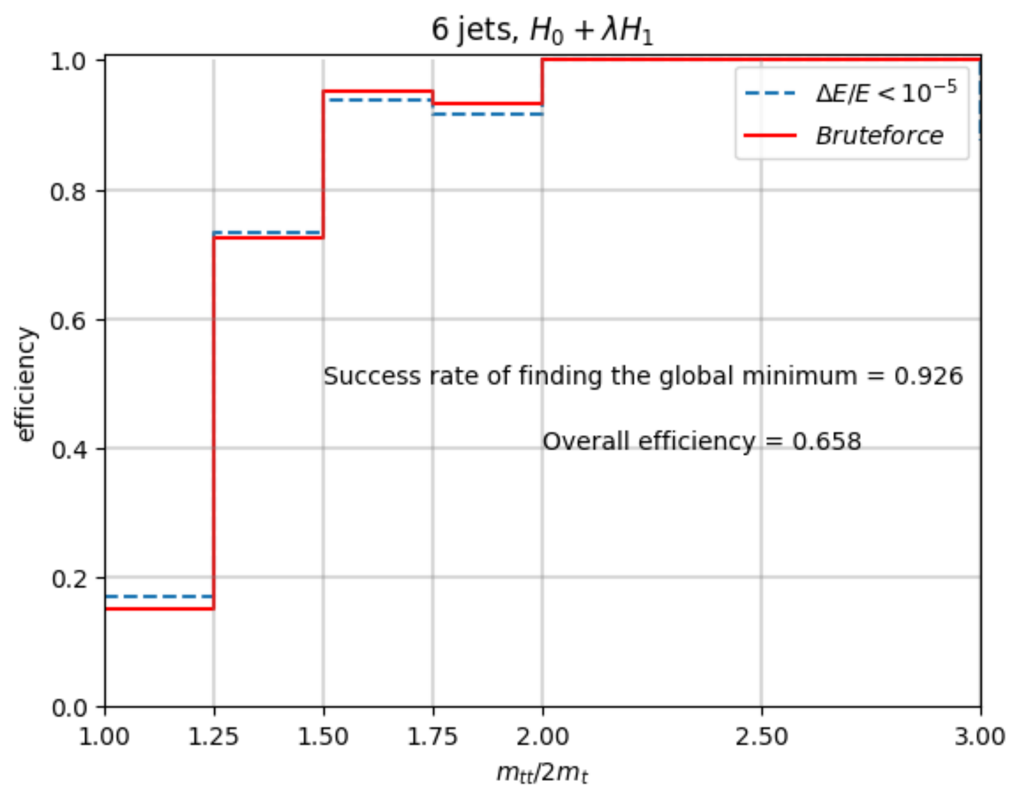
15 parameters
no classical optimizers needed



Performant near-term quantum combinatorial optimization : 2404.16135

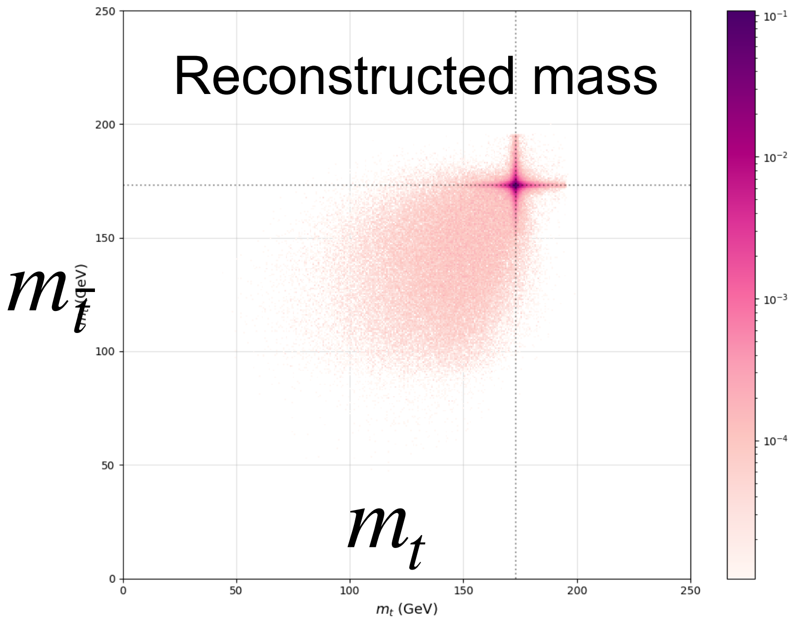


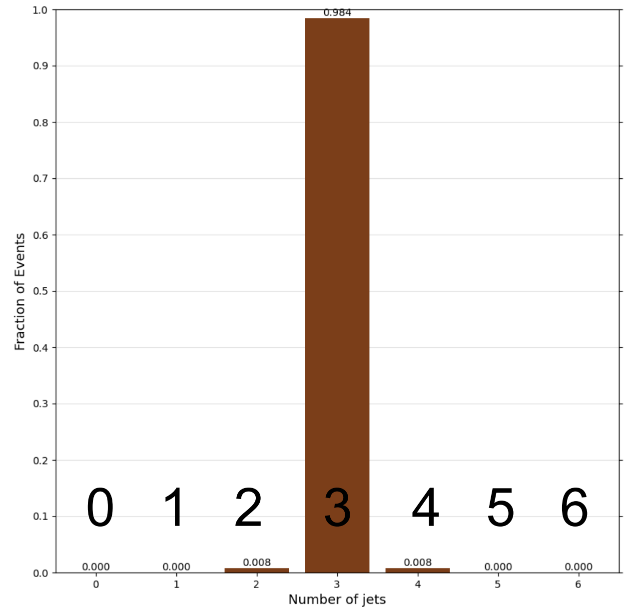
Success probability of finding the global minimum



Efficiency of resolving the combinatorial problem

Parton-level truth and hemisphere method





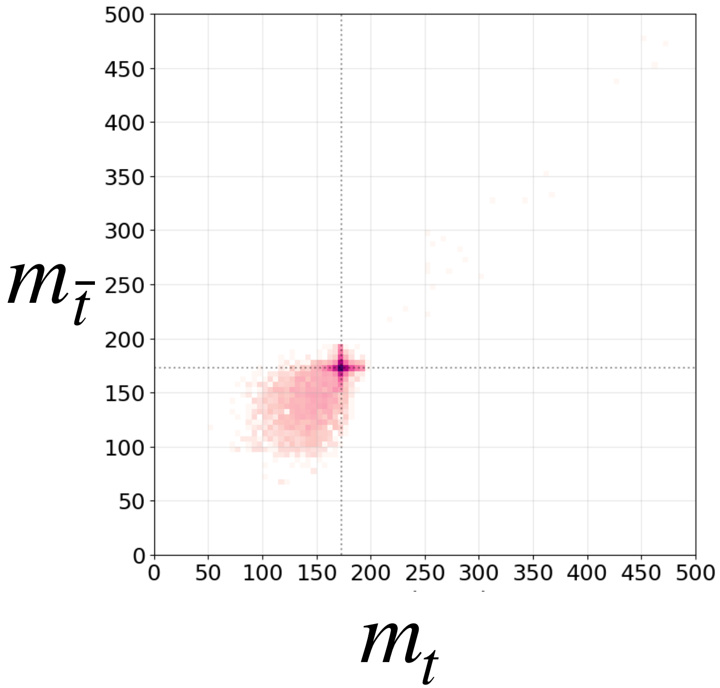
Jet assignment

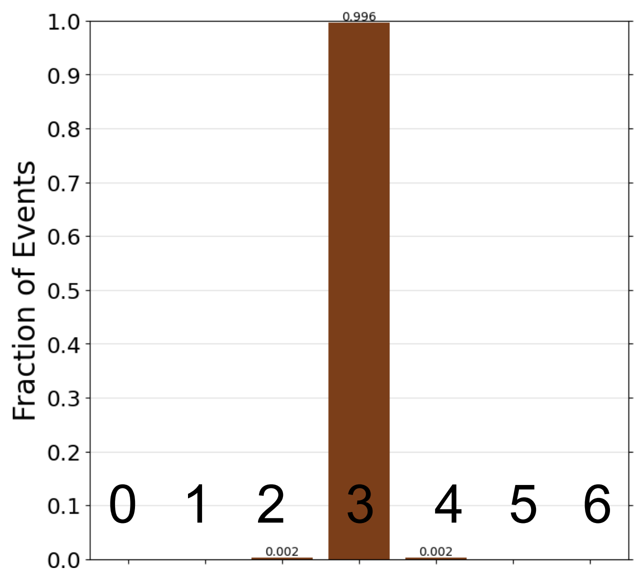
parton-level truth

79% overal efficiency

No mass information used.

Hybrid quantum-classical approach for combinatorial problems at hadron colliders: 2410.22417





Jet assignment

VarQITE method

79% overal efficiency

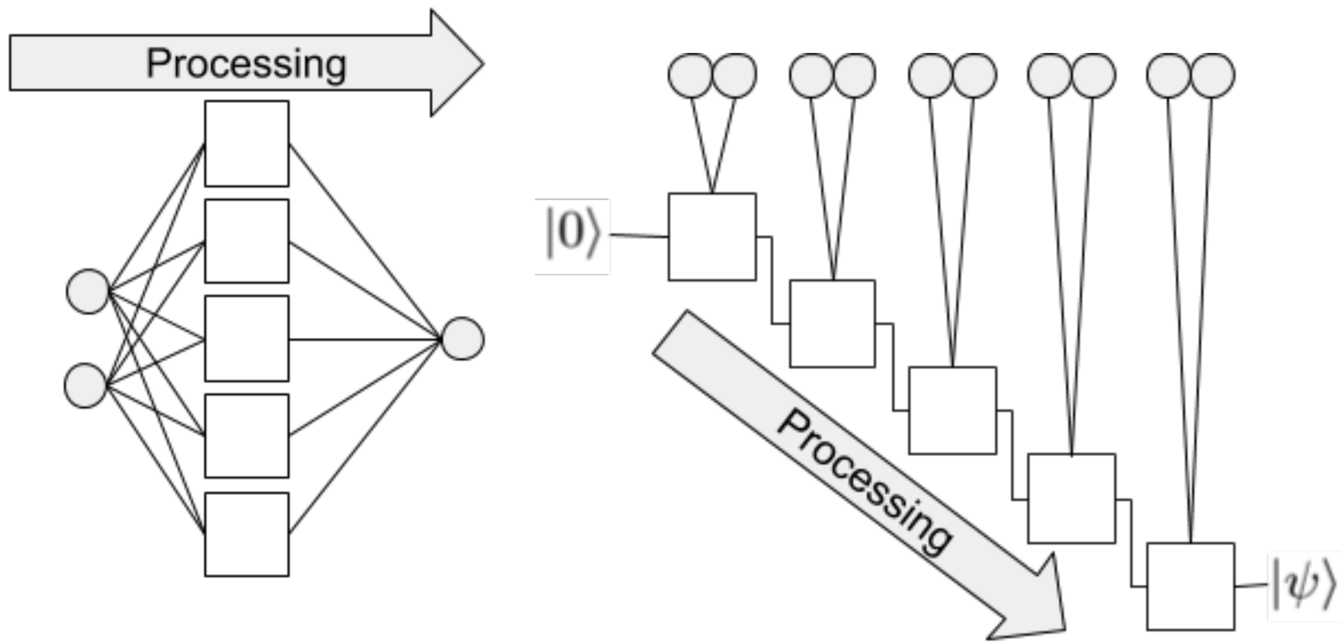
No mass information used.

Summary

- This field combines physics, math, computer science, and engineering — so whatever you're into, you can be part of it.
- Quantum computers could revolutionize many areas such as security, medicine, materials, AI, climate modeling, and more.
  - Quantum biology, quantum finance, post-quantum security,
- It's a young field — we need your ideas, your questions, and your creativity.
- What You Can Do Now:
  - Get curious: try a quantum simulator (like IBM Quantum Experience).
  - Learn linear algebra and probability — they're the language of quantum.
  - Ask "what if?" — because today's science fiction is tomorrow's technology.

Data re-uploading for a universal quantum classifier

1907.02085



(a) Neural network (b) Quantum classifier

- Universal approximation theorem
- We can approximate a function $F(\vec{x})$ with $f(\vec{x}, \vec{\theta})$, where $\vec{x}$ is an input feature and $\vec{\theta}$ is a learnable parameter.
- The cost function (ex. MSE) to be minimized is $\sum_{i=1}^{n} | F(\vec{x}_i) - f(\vec{x}_i, \vec{\theta}) |^2$

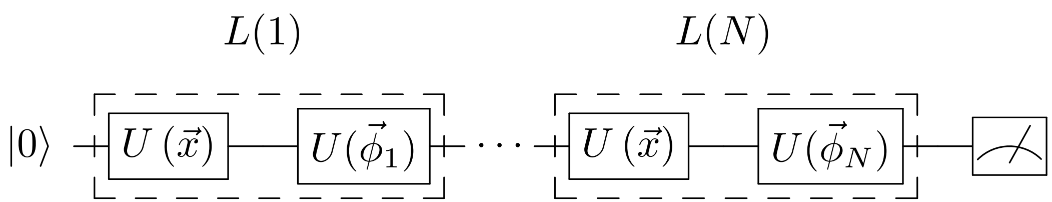
Single qubit classifier using data re-uploading

1907.02085

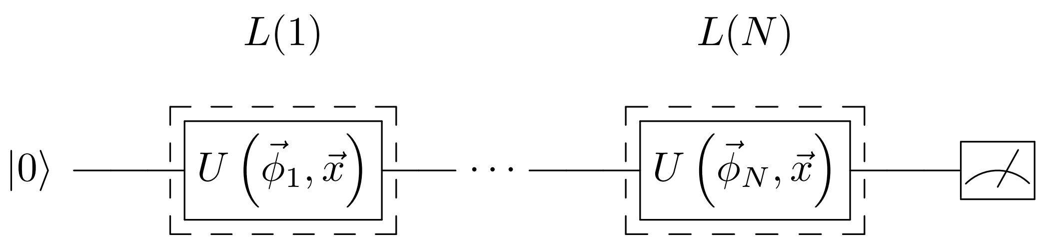
- Consider the three dimensional data, $\vec{x}$. (can be generalized.)
- Date can be re-uploaded using unitary transformation $U(\vec{x})$ rotating the qubit.
- The single-qubit classifier has the following structure: $|\psi\rangle = \mathcal{U}(\vec{\phi}, \vec{x})|0\rangle$

$$\mathcal{U}(\vec{\phi}, \vec{x}) = L(N) \dots L(1) \qquad L(i) \equiv U(\vec{\phi}_i)U(\vec{x}) \qquad \vec{\phi} = (\phi_1, \phi_2, \phi_3)$$

$$\mathcal{U}(\vec{\phi}, \vec{x}) \equiv U(\vec{\phi}_N)U(\vec{x}) \dots U(\vec{\phi}_1)U(\vec{x}) \qquad U(\phi_1, \phi_2, \phi_3) \in SU(2)$$



(a) Original scheme



(b) Compressed scheme

$$L(i) = U\left(\vec{\theta}_i + \vec{w}_i \circ \vec{x}\right)$$

Hadamard product of $\vec{w}_i$ and $\vec{x}$:

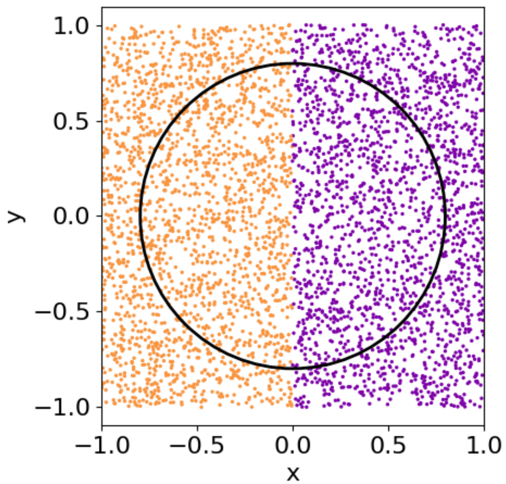
$$\vec{w}_i \circ \vec{x} = \left(w_i^1 x^1, w_i^2 x^2, w_i^3 x^3\right)$$

$$L(i) = U\left(\vec{\theta}_i^{(k)} + \vec{w}_i^{(k)} \circ \vec{x}^{(k)}\right) \cdots U\left(\vec{\theta}_i^{(1)} + \vec{w}_i^{(1)} \circ \vec{x}^{(1)}\right)$$

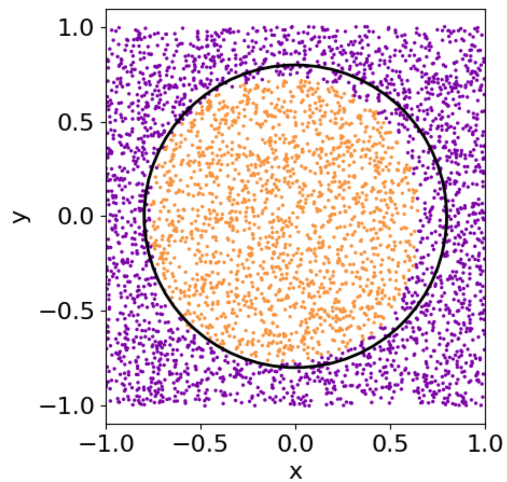
The number of parameters = 3 N

$$U(\vec{\phi}) = U(\phi_1, \phi_2, \phi_3) = e^{i\phi_2\sigma_z} e^{i\phi_1\sigma_y} e^{i\phi_3\sigma_z} \quad \text{or} \quad U(\vec{\phi}) = e^{i\vec{\omega}(\vec{\phi})\cdot\vec{\sigma}}$$

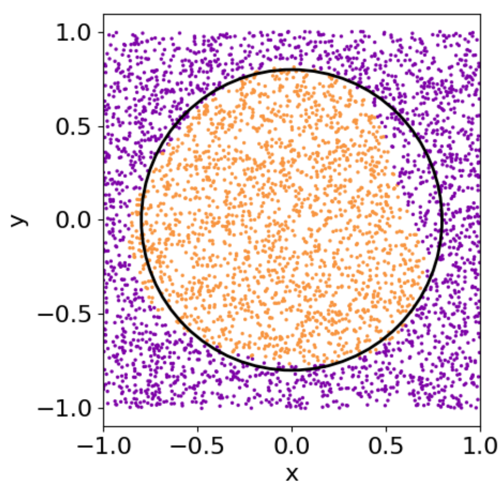
Example: binary classification



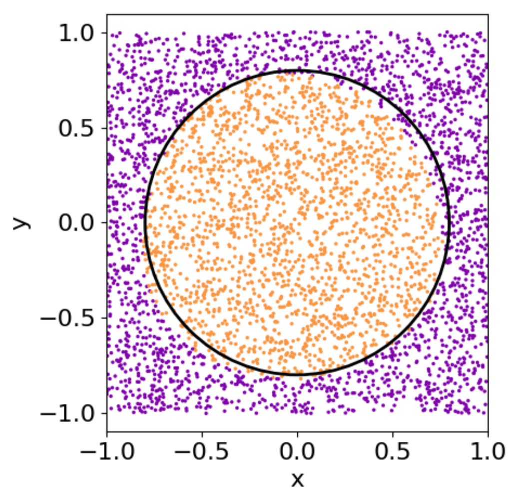
(a) 1 layer



(b) 2 layers



(c) 4 layers



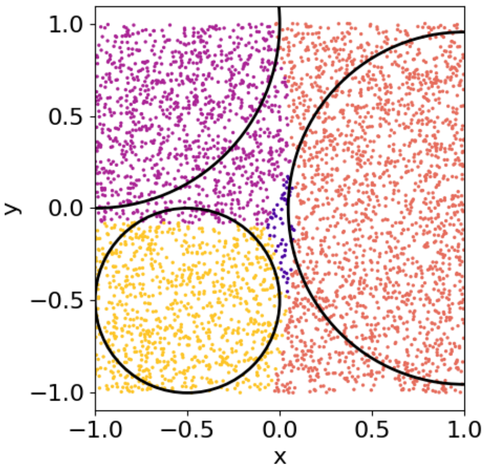
(d) 8 layers

	χ_f^2		
Qubits	1	2	
Layers		No Ent.	Ent.
1	0.50	0.75	–
2	0.85	0.80	0.73
3	0.85	0.81	0.93
4	0.90	0.87	0.87
5	0.89	0.90	0.93
6	0.92	0.92	0.90
8	0.93	0.93	0.96
10	0.95	0.94	0.96

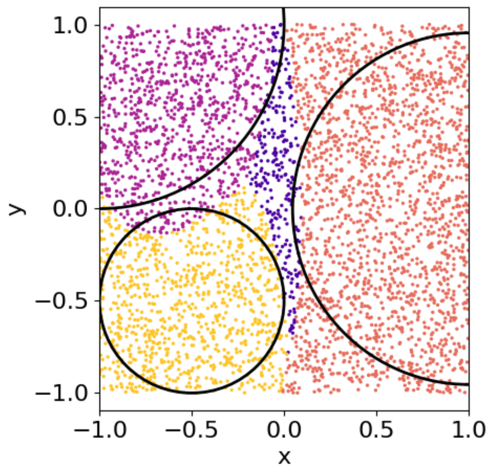
The number of parameters = 3 N

1907.02085

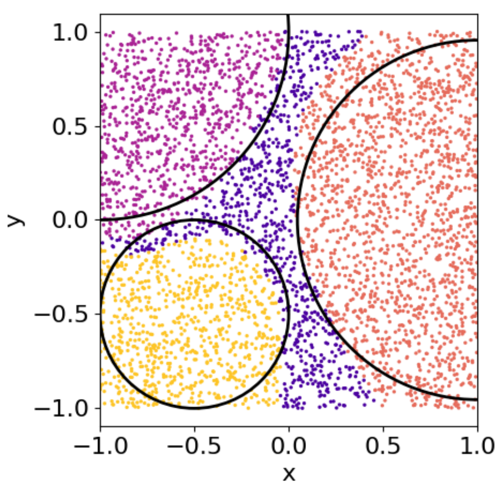
Example: 4 classes



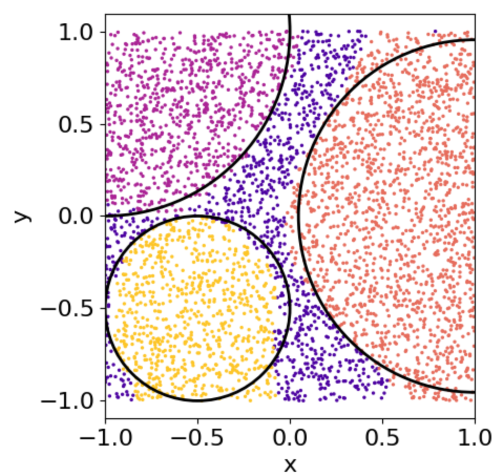
(a) 1 layer



(b) 3 layers



(c) 4 layers

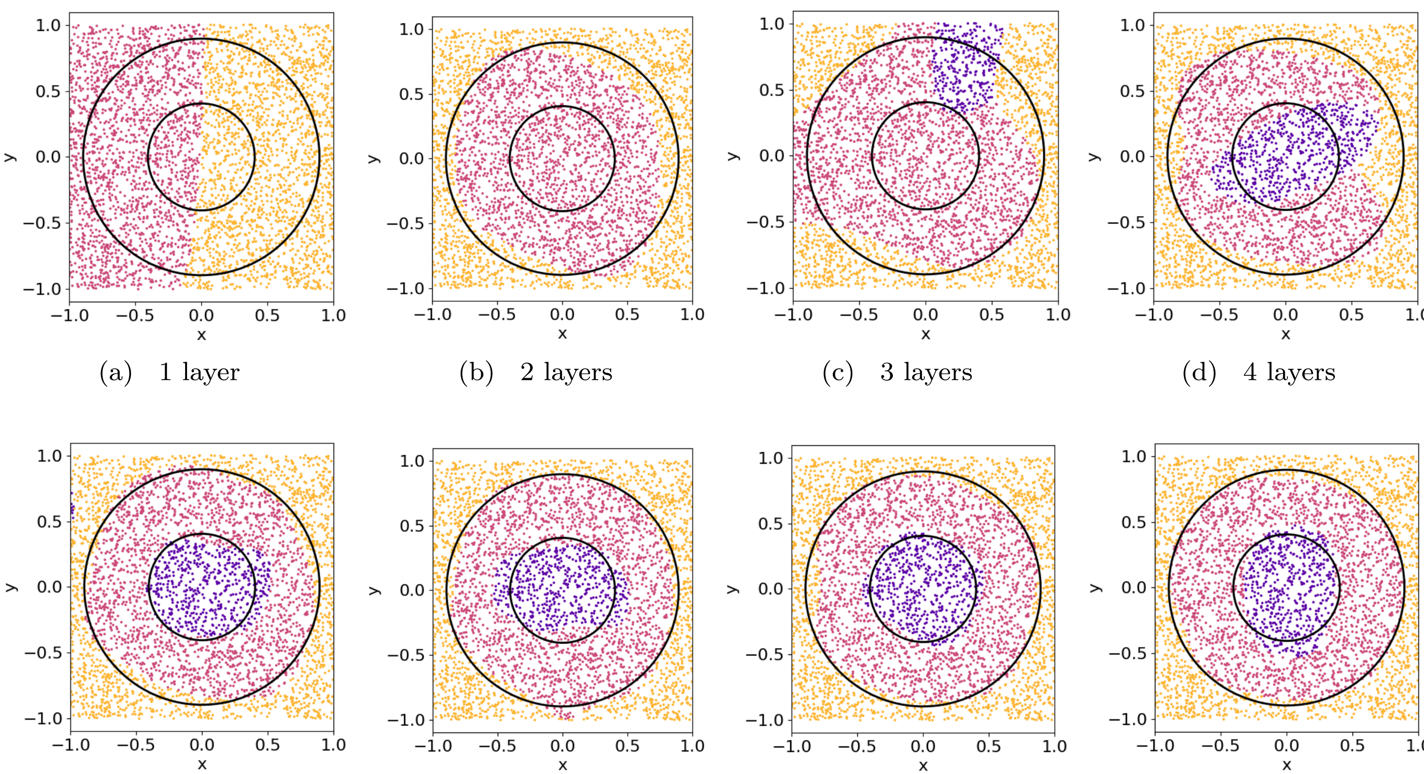


(d) 10 layers

	χ_f^2		
Qubits	1	2	
Layers		No Ent.	Ent.
1	0.73	0.56	–
2	0.79	0.77	0.78
3	0.79	0.76	0.75
4	0.84	0.80	0.80
5	0.87	0.84	0.81
6	0.90	0.88	0.86
8	0.89	0.85	0.89
10	0.91	0.86	0.90

1907.02085

Example: 3 classes



(a) 1 layer (b) 2 layers (c) 3 layers (d) 4 layers

(e) 5 layers (f) 6 layers (g) 8 layers (h) 10 layers

	χ^2_f		
Qubits	1	2	
Layers		No Ent.	Ent.
1	0.34	0.51	–
2	0.57	0.63	0.59
3	0.80	0.68	0.65
4	0.84	0.78	0.89
5	0.92	0.86	0.82
6	0.93	0.91	0.93
8	0.90	0.89	0.90
10	0.90	0.91	0.92

1907.02085

Single qubit classifier: example

Problem	Classical classifiers NN	SVC	Quantum classifier χ^2_f	χ^2_{wf}
Circle	0.96	0.97	0.96	0.97
3 circles	0.88	0.66	0.91	0.91
Hypersphere	0.98	0.95	0.91	0.98
Annulus	0.96	0.77	0.93	0.97
Non-Convex	0.99	0.77	0.96	0.98
Binary annulus	0.94	0.79	0.95	0.97
Sphere	0.97	0.95	0.93	0.96
Squares	0.98	0.96	0.99	0.95
Wavy Lines	0.95	0.82	0.93	0.94

Comparison between single-qubit quantum classifier and two well-known classical classification techniques: a neural network (NN) with a single hidden layer composed of 100 neurons and a support vector classifier (SVC), both with the default parameters as defined in scikit-learn python package. This table shows the best success rate, being 1 the perfect classification, obtained after running ten times the NN and SVC algorithms and the best results obtained with single-qubit classifiers up to 10 layers.

1907.02085

Quantum Annealing (Gradient-Free quantum optimization)

- H_p is the problem Hamiltonian whose ground state encodes the solution to the optimization problem
- H_0 is the initial Hamiltonian whose ground state is easy to prepare.
- Prepare a quantum system to be in the ground state of H_0 and evolve the system using the following time-dependent Hamiltonian,

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_p$$

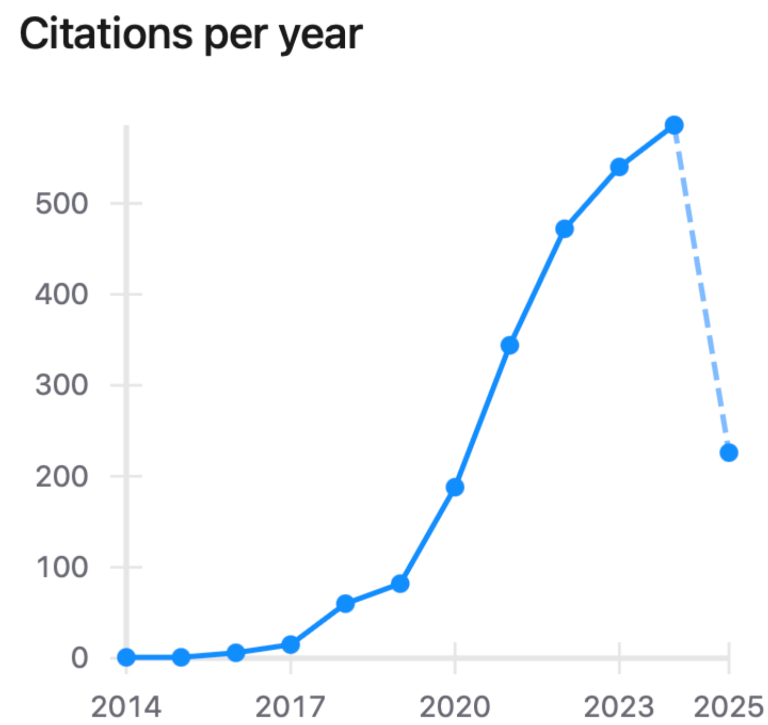
- The system will remain to its ground state at all times for a large T, which means for t=T, the system will be in the ground state of H_p, our problem Hamiltonian.
- D-wave has built Quantum Annealing that solves optimization problem by transferring the original optimization to a hardware, that allows nearest neighbor interaction of qubits.
- Compared to the processing time of O(2^n) with the simplest but a robust brute-force scanning algorithm with a classical computer, a quantum annealer can have an enormous advantage in the computation complexity as $T_{\mathrm{QUBO}}(n) \sim \mathcal{O}(n^2) \ll \mathcal{O}(2^n)$

Apolloni, Bianchi, De Falco 1988

Quantum Approximate Optimization Algorithm (QAOA)

- Abstract: We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on a positive integer p and the quality of the approximation improves as p is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing. If p grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed p. For p = 1, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.6924 times the size of the optimal cut.

1411.4028, E. Farhi, J. Goldstone, S. Gutmann



Experimental realization

PRL **110,** 230501 (2013) PHYSICAL REVIEW LETTERS week ending 7 JUNE 2013

Experimental Quantum Computing to Solve Systems of Linear Equations

X.-D. Cai,[1] C. Weedbrook,[2] Z.-E. Su,[1] M.-C. Chen,[1] Mile Gu,[3,4] M.-J. Zhu,[1] Li Li,[1,*] Nai-Le Liu,[1,†] Chao-Yang Lu,[1,‡] and Jian-Wei Pan[1]

[1]*Hefei National Laboratory for Physical Sciences at Microscale and Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230026, China*

[2]*Center for Quantum Information and Quantum Control, Department of Electrical and Computer Engineering, and Department of Physics, University of Toronto, Toronto M5S 3G4, Canada*

[3]*Centre for Quantum Technologies, National University of Singapore, Singapore 117543*

[4]*Center for Quantum Information, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China*

(Received 6 March 2013; published 6 June 2013)

Solving linear systems of equations is ubiquitous in all areas of science and engineering. With rapidly growing data sets, such a task can be intractable for classical computers, as the best known classical algorithms require a time proportional to the number of variables N. A recently proposed quantum algorithm shows that quantum computers could solve linear systems in a time scale of order $\log(N)$, giving an exponential speedup over classical computers. Here we realize the simplest instance of this algorithm, solving 2×2 linear equations for various input vectors on a quantum computer. We use four quantum bits and four controlled logic gates to implement every subroutine required, demonstrating the working principle of this algorithm.

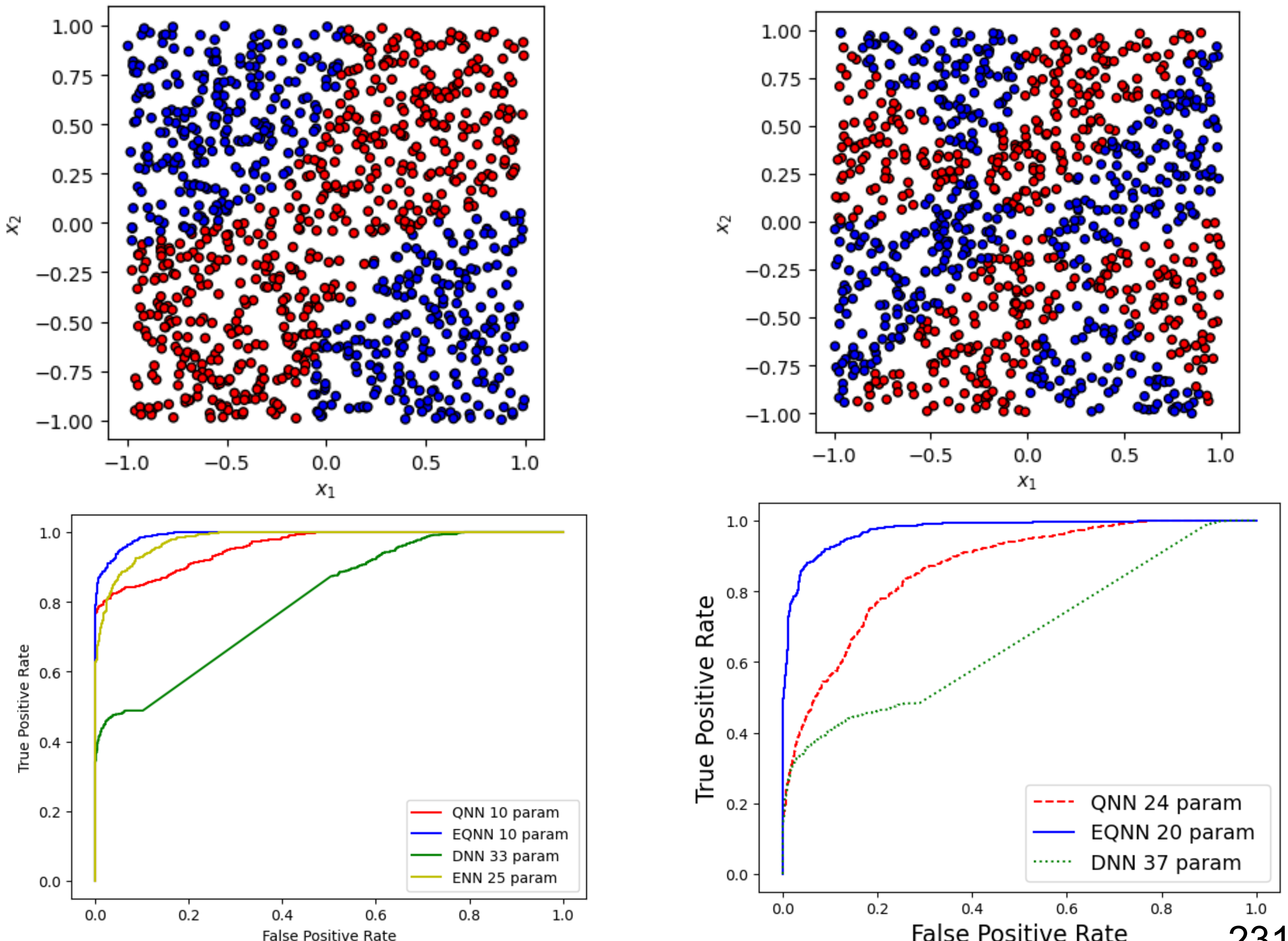
"A two-qubit photonic quantum processor and its application to solving systems of linear equations". Scientific Reports. 4: 6115.
"Experimental realization of quantum algorithm for solving linear systems of equations". Physical Review A. 89 (2): 022313

Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing, Phys.Rev.Lett. 122 (2019) 6, 060504
===> "Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing". Phys. Rev. A 99, 012320. 8 dimensional linear equation.

P vs NP

- In Theoretical Computer Science, the two most basic classes of problems are P and NP.
- P includes all problems that can be solved “efficiently”.
  - For example: add two numbers. The formal definition of "efficiently" is in time that's polynomial in the input's size.
- NP (nondeterministic polynomial (time)) includes all problems that given a solution, one can efficiently verify that the solution is correct.
  - An example is the following problem: given a bunch of numbers, can they be split into two groups such that the sum of one group is the same as the other. Clearly, if one is given a solution (two groups of numbers), it's simple to verify that the sum is the same. (This is a partitioning problem).
- What's unknown is whether problems such as the one above have an efficient algorithm for finding the solution. This is the (in)famous (unsolved) P = NP problem, and the common conjecture is that no such algorithm exists.
- Now, NP hard problems are such problems that were shown that if they can be efficiently solved (which, as mentioned, is believed to not be the case), then each and every problem in NP (each and every problem whose results can be efficiently verified) can be efficiently solved. In other words, if you're up to showing that P=NP, you might want to take a stand at any of those NP-hard problems since they are "equivalent" in some way to all others.

Equivariant Quantum Neural Networks



2311.18744

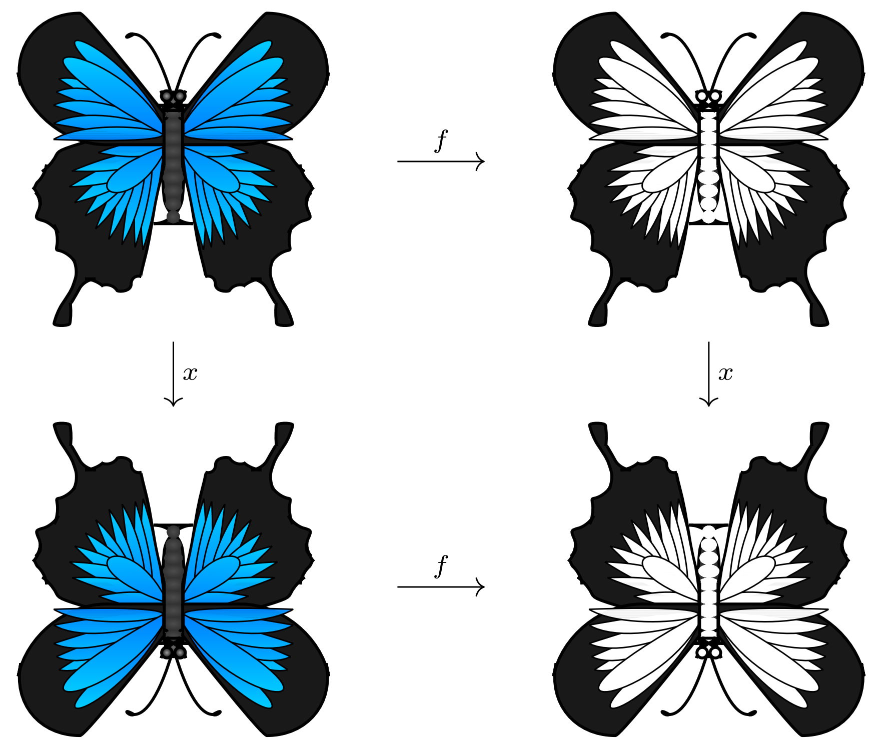
2205.0736

Let $\mathbb{V}$ and $\mathbb{W}$ be sets, and $f\colon \mathbb{V} \to \mathbb{W}$ a function. If a group G acts on both $\mathbb{V}$ and $\mathbb{W}$, and this action commutes with the function f:

$$f(x \cdot v) = x \cdot f(v) \quad \text{for all } v \in \mathbb{V},\ x \in G,$$

then we say that f is G-*equivariant*. The special case where G acts trivially on $\mathbb{W}$ is called G-*invariant*.

$$f(x \cdot v) = x \cdot f(v) \text{ for all } v \in \mathbb{V}$$



Let $\mathbb{V} = \mathbb{W}$ be the set of all images. Let the group $G = \{1, x\} \cong \mathbb{Z}/2\mathbb{Z}$ act on $\mathbb{V}$ via *top-bottom* reflection, i.e., $x \cdot v$ is the image whose value at (p_1, p_2) is $v(p_1, -p_2)$. Let $\sigma\colon \mathbb{R}^3 \to \mathbb{R}^3$,

$$\sigma(r, g, b) = \begin{cases} (0,0,0) & \text{if } r = g = b = 0, \\ (255, 255, 255) & \text{otherwise.} \end{cases}$$

Here $(0,0,0)$ and $(255,255,255)$ are the RGB encodings for pitch black and pure white respectively. So the map $f\colon \mathbb{V} \to \mathbb{V}$, $f(v) = \sigma \circ v$ transforms a color image into a black-and-white image.

Qubits and Pauli's matrices

$$\sigma_1 = \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\sigma_2 = \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\sigma_3 = \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$[\sigma_i, \sigma_j] \equiv \sigma_i \sigma_j - \sigma_j \sigma_i = 2i\epsilon_{ijk}\sigma_k$$

$$\{\sigma_i, \sigma_j\} \equiv \sigma_i \sigma_j + \sigma_j \sigma_i = 2\delta_{ij}$$

$$\sigma_i \sigma_j = 2\delta_{ij} + i\,\epsilon_{ijk}\,\sigma_k$$

- Qubit $(\,|\psi\rangle, |\psi\rangle) \equiv \langle\psi|\psi\rangle = 1)$:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle = \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{pmatrix}$$

- Conjugate (dual vector or bra-vector):

$$\langle\psi| = \cos\frac{\theta}{2}\langle 0| + e^{-i\phi}\sin\frac{\theta}{2}\langle 1| = \begin{pmatrix} \cos\frac{\theta}{2} & e^{-i\phi}\sin\frac{\theta}{2} \end{pmatrix}$$

- A set of all $|\psi\rangle$ (ket-vector) forms a vector space (Hilbert space)
- Pauli's matrices are generators of rotations in two dimensional complex plane.

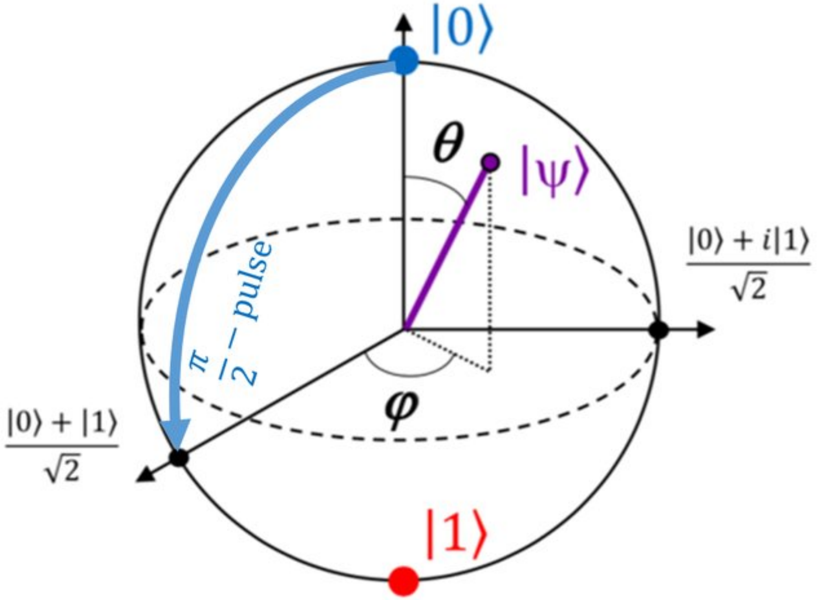
$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$ Computational basis

$$|\pm\rangle \equiv \frac{|0\rangle \pm |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ \pm 1 \end{pmatrix}$$ Hadamard basis

$$R(\vec{\theta}) = \exp\left(-i\frac{\vec{\theta}\cdot\vec{\sigma}}{2}\right)$$

Bloch Sphere

- Each (normalized) state of the qubit can be uniquely associated with a point on the unit sphere.



$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

$$|\psi\rangle \;\longleftrightarrow\; (\theta, \phi) \;\longleftrightarrow \hat{r} = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}$$

$$|0\rangle : \;\; \theta = 0\,, \phi = \text{arbitrary} \longrightarrow \hat{r} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$|1\rangle : \;\; \theta = \pi\,, \phi = \text{arbitrary} \longrightarrow \hat{r} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

$$|+\rangle : \;\; \theta = \pi/2\,, \phi = 0 \longrightarrow \hat{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|+i\rangle : \;\; \theta = \pi/2\,, \phi = \pi/2 \longrightarrow \hat{r} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|-\rangle : \;\; \theta = \pi/2\,, \phi = \pi \longrightarrow \hat{r} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

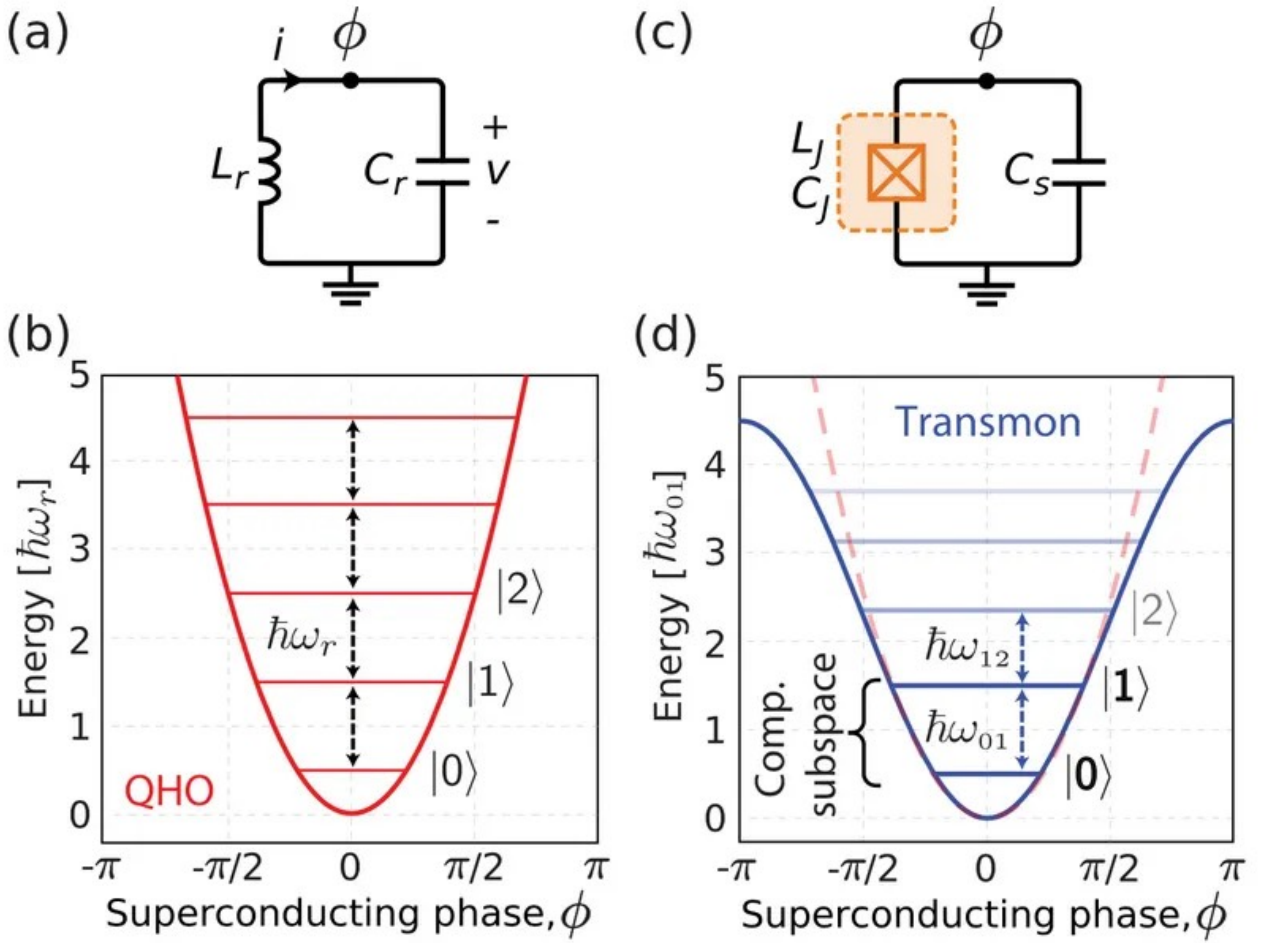
$$|+-\rangle : \;\; \theta = \pi/2\,, \phi = 3\pi/2 \longrightarrow \hat{r} = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

Quantum Measurements

- When we perform a measurement onto an orthogonal basis, the qubit collapses to one of the basis states with probability given by the corresponding amplitude. For example, if we perform a projection onto z-axis (the states $|0\rangle$ and $|1\rangle$), we get $P(0) = |\langle 0|\psi\rangle|^2 = \cos^2\frac{\theta}{2}$ and $P(1) = |\langle 1|\psi\rangle|^2 = \sin^2\frac{\theta}{2}$.

- Born rule: the probability that a state $|\psi\rangle$ collapses during a projective measurement onto a basis $\{|x\rangle, |x^\perp\rangle\}$ is given by $P(x) = |\langle x|\psi\rangle|^2$ and $P(x^\perp) = |\langle x^\perp|\psi\rangle|^2$.

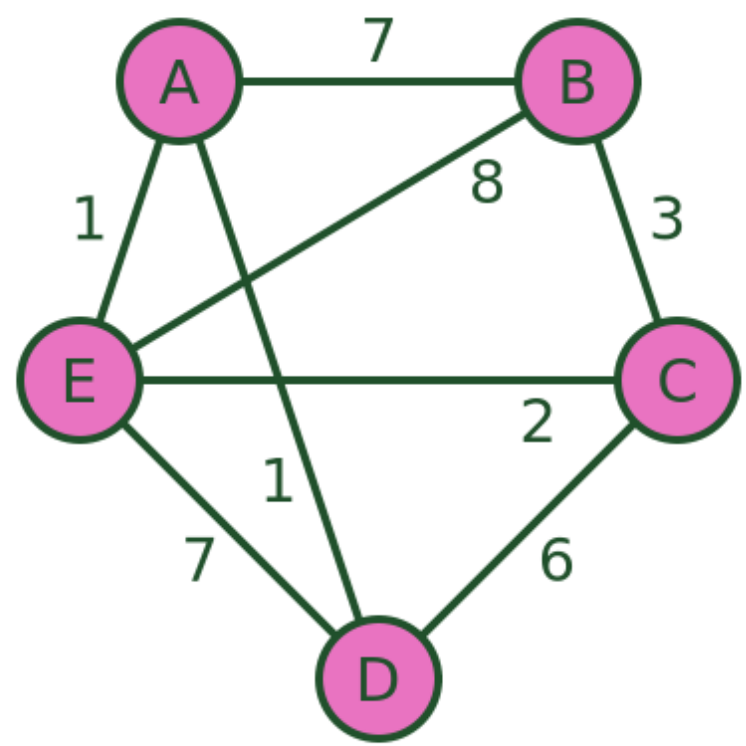
$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle = \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{pmatrix}$$



Transmon qubit: Capacitor + Josephson junction (JJ)

Traveling salesman problem (TSP)

- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- It is an NP-hard problem in combinatorial optimization, important in theoretical computer science and operations research.



Traveling salesman problem (TSP)

- Performance of the Bellman–Held–Karp algorithm
- Time Complexity
  - The Bellman-Held-Karp algorithm has a time complexity of $\mathcal{O}(n^2 2^n)$, where n is the number of cities.
  - This means the runtime increases exponentially with the number of cities, making it impractical for very large instances of the TSP.
  - However, this is still significantly better than the brute-force approach that has a time complexity of $\mathcal{O}(n!)$, where 'n' is the number of cities.
- Space Complexity
  - The algorithm has a space complexity of $\mathcal{O}(n2^n)$.
  - This is due to the algorithm needing to store intermediate results in a table, especially for storing the shortest path costs for all possible subsets of cities and destination cities.
  - This space requirement can become a limiting factor for larger problems as well.
- Query complexity of QAOA on TSP: $\mathcal{O}(\sqrt{n!})$. $(n^2 2^n \gg \sqrt{n!}$ for a large $n)$
  - Challenges
    - Implementing QAOA on real quantum hardware
    - Optimization of Parameters:

Ising formulations of many NP problems, 1302.5843

Traveling salesman problem (TSP)

- Label vertices $1, \cdots, N$ and the edge set (uv) to be directed. (For undirected, add (uv) and (vu)).
- Introduce N^2 bits $x_{v,i}$, where v represents the vertex and i represents its order to the edge set.
- Every vertex can only appear once in a cycle, and that there must be a j-th node in the cycle for each j.
- For the nodes in our prospective ordering, if $x_{u,j}$ and $x_{u,j+1}$ are both 1, then there should be an energy penalty if $(uv) \notin E$.

$$H_A = A \sum_{v=1}^{n} \left(1 - \sum_{j=1}^{N} x_{v,j}\right)^2 + A \sum_{j=1}^{n} \left(1 - \sum_{v=1}^{N} x_{v,j}\right)^2 + A \sum_{(uv)\notin E} \sum_{j=1}^{N} x_{u,j} x_{v,j+1}$$

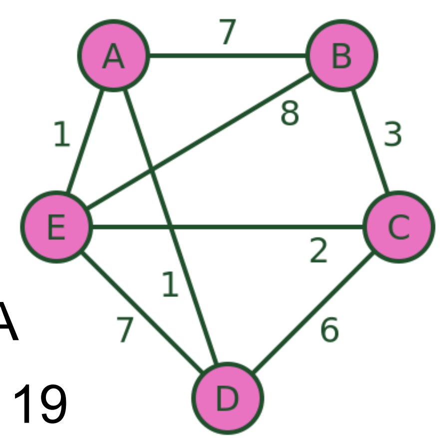
	1	2	3	4	5
A	1				
B			1		
C				1	
D					
E		1			1

- $A > 0$ is a constant. The ground state of this system has $H = 0$, only if we have an ordering of vertices where each vertex is only included once, and adjacent vertices in the cycle have edges on the graph – i.e., we have a Hamiltonian cycle.
- The traveling salesman also needs to minimize the total traveling distance (weight)

$$H_B = B \sum_{(uv)\in E} W_{uv} \sum_{j=1}^{N} x_{u,j} x_{v,j+1}$$

- B should be small enough that it is never favorable to violate the constraints of H_A. $\quad 0 < B \max(W_{uv}) < A$

$$H_{\text{full}} = H_P = H_A + H_B$$



AEBCDA

1+8+3+6+1 = 19

Portfolio optimization

Markowitz model

risk minus returns

no. of assets of type-i

risk tolerance

$$F = \sum_{ij} C_{ij} n_i n_j - \zeta \sum_i R_i n_i$$

correlations between assets

return on i'th asset

Spin glasses

Sherrington-Kirkpatrick model

Hamiltonian

spin at site i

$$H = \sum_{ij} J_{ij} s_i s_j + \sum_i h_i s_i$$

couplings between sites

external magnetic field at i'th site

Neural networks

Hopfield model

energy function

state of neuron i

$$E = \sum_{ij} w_{ij} v_i v_j + \sum_i \theta_i v_i$$

weights of the learning rule

activation threshold for i'th neuron

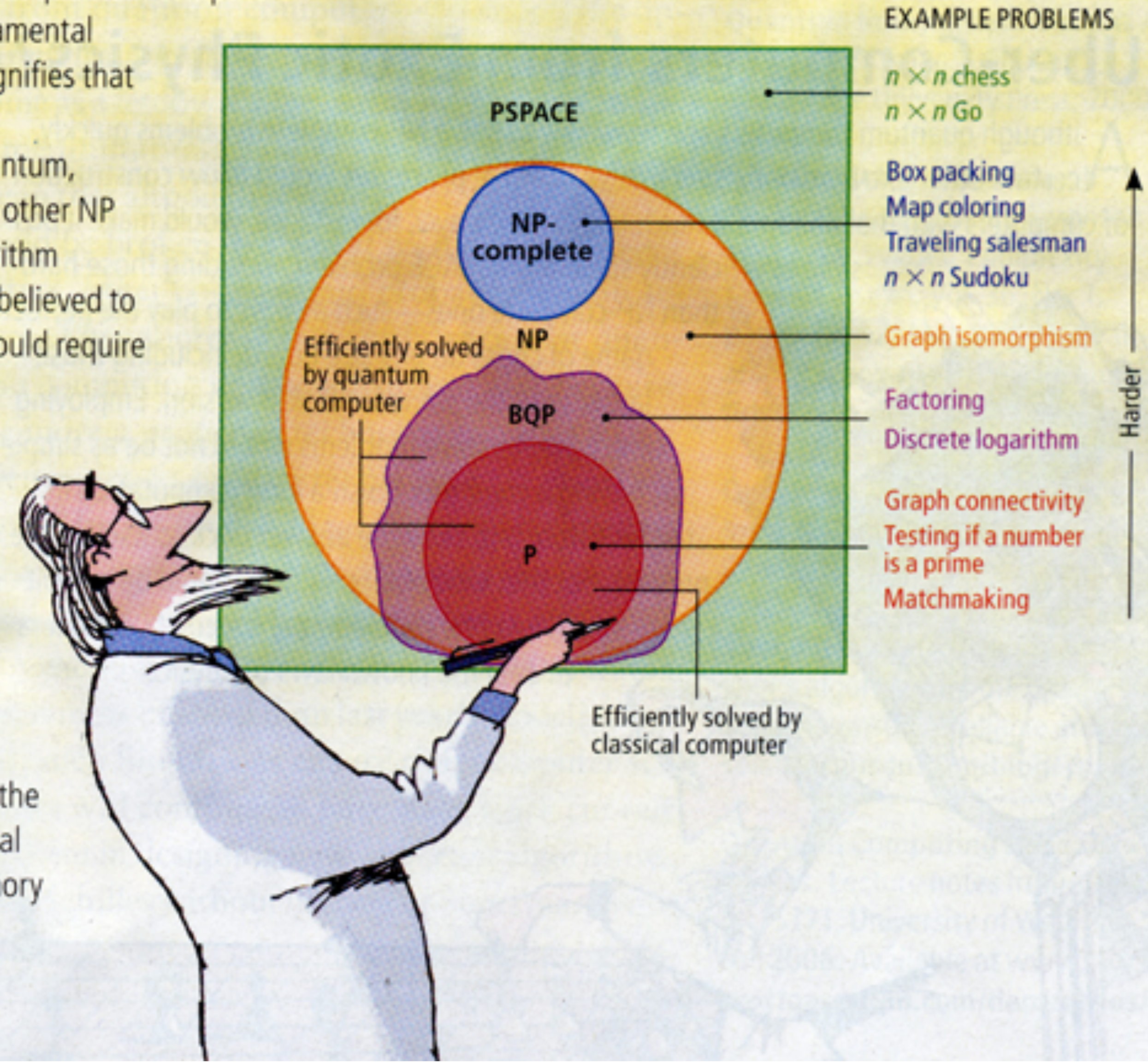
Where Quantum Computers Fit In

The map at the right depicts how the class of problems that quantum computers would solve efficiently (BQP) might relate to other fundamental classes of computational problems. (The irregular border signifies that BQP does not seem to fit neatly with the other classes.)

The BQP class (the letters stand for *bounded-error, quantum, polynomial time*) includes all the P problems and also a few other NP problems, such as factoring and the so-called discrete logarithm problem. Most other NP and all NP-complete problems are believed to be outside BQP, meaning that even a quantum computer would require more than a polynomial number of steps to solve them.

In addition, BQP might protrude beyond NP, meaning that quantum computers could solve certain problems faster than classical computers could even check the answer. (Recall that a conventional computer can efficiently verify the answer of an NP problem but can efficiently solve only the P problems.) To date, however, no convincing example of such a problem is known.

Computer scientists do know that BQP cannot extend outside the class known as PSPACE, which also contains all the NP problems. PSPACE problems are those that a conventional computer can solve using only a polynomial amount of memory but possibly requiring an exponential number of steps.



Scott Aaronson, The Limits of Quantum Computers, Scientific American Magazine 298(3), 62-69 (2008)

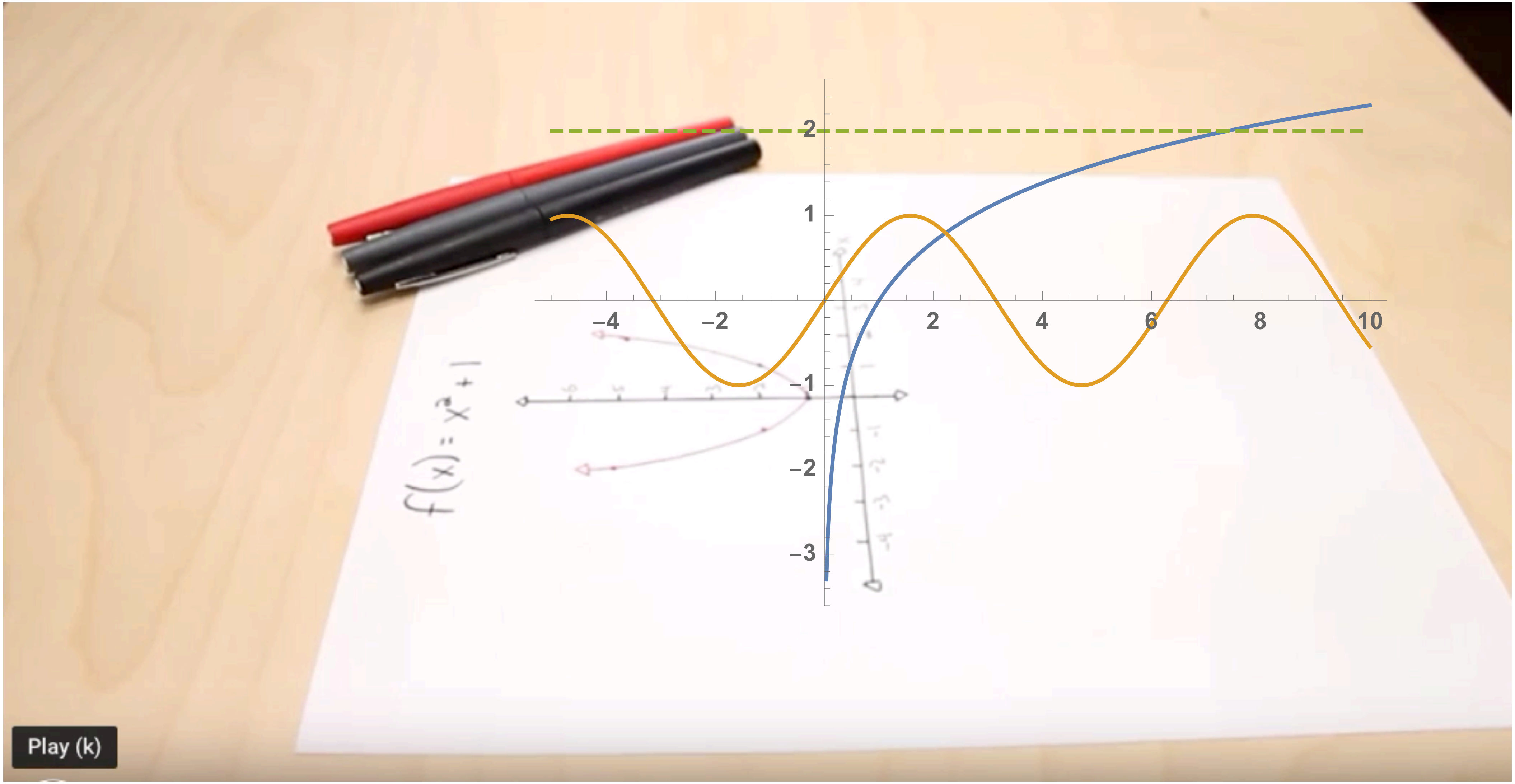
The exponential Speed-Up

Description	Classical Algorithm	Quantum Algorithm	Reference
• **Grover's algorithm:** ✓ searches an unstructured database (or an unordered list) with N entries	$O(N)$	$O(\sqrt{N})$	L.K. Grover, "A fast quantum mechanical algorithm for database search." Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. ACM (1996).
• **Shor's algorithm:** ✓ Integer factorization	$O\left(e^{1.9(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}}\right)$	$O\left((\log N)^2(\log\log N)(\log\log\log N)\right)$	D. Beckman, et al. "Efficient networks for quantum factoring." Physical Review A 54.2 (1996).
• **Quantum Fourier Transform:** ✓ Discrete Fourier transform of size *N*	$O(N \log N)$	$O(\log N \ \log\log N)$	P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proc. 35th Annual Symp. on Foundations of Comp. Sci., pp.124-134(1994)
• **Eigenvalue solver:** ✓ To find eigenvalues and eigenvectors of a local Hamiltonian	$O(N^{2\sim 2.236})$	$O\left((\log N)^4\right)$	Abrams and S. Lloyd, "Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectorsg." Phys. Rev. Lett. 83, 24, p.5162 (1999).
• **Matrix inversion:** ✓ Finding inverse matrix. This can be applied to find a vector x satisfying Ax=b, where A and b are Hermitian N × N matrix and a unit vector, respectively	$O(Ns\kappa(\log 1/\epsilon))$	$\tilde{O}\left(\frac{\log N\, s^2\kappa^2}{\epsilon}\right)$	A.W. Harrow, Avinatan Hassidim, and S. Lloyd. "Quantum algorithm for linear systems of equations." Physical review letters 103.15 (2009): 150502.
• **Distance (inner product) evaluation:** ✓ Calculating inner product between a given N dimensional vector and each N dimensional vector of M samples	$O(MN)$	$O(\log NM)$	S. Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning." arXiv preprint arXiv:1307.0411 (2013).

The exponential Speed-Up

Learning Problem	Classical Algorithm	Quantum Algorithm	Reference
• **k-means problem:** ✓ Assigning M vectors to k clusters in a way that minimizes the average distance to the centroid of the cluster	$O(M^2N)$	$O(M\log(MN))$	Lloyd, Seth, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning." arXiv preprint arXiv:1307.0411 (2013).
• **Principle component analysis (PCA) problem:** ✓ To convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated	$O(d^2 \log R + d R^2)$	$O(R \log d)$	Lloyd, Seth, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis." Nature Physics 10.9 (2014): 631-633.
• **Support vector machine (SVM) problem:** ✓ To classify data clusters with support vector learning	$O(NM)$	$O(\log(NM))$	P. Rebentrost, M. Mohseni, S. Lloyd, "Quantum Support Vector Machine for Big Data Classification," PRL 113, p.130503, 2014.
• **Quantum Neural Network (QNN) problem:** ✓ Qubit (or Node) requirement for neural network machine learning	$O(ND)$	$O(\log(N))$	S. Gupta and R.K.P. Zia, "Quantum Neural Network," Journal of Computer and System Sciences 63, 355–383 (2001).
• **Classification Problem:** ✓ Instant measure of hamming distance among training vector data and query vector	$O(M^3)$	$O(1)$?	M. Schuld, M. Fingerhuth, and F. Petruccione, "Implementing a distance-based classifier with a quantum interference circuit," EPL, v119,n6, 60002,2017
• **Learning parity with noise (LPN) problem:** ✓ For given some samples (x, f(x)), estimating the function f computing the parity of bits at some fixed locations	N queries in a noiseless channel	O(logN) queries in a noisy (depolarizing) channel	A.W. Cross, S. Graeme, and J.A. Smolin. "Quantum learning robust against noise." Physical Review A 92.1 (2015): 012327.

Q1: solve for “x”



$$y = x^2 + 1$$

from Welch Labs

USC Study Demonstrates Unconditional Exponential Quantum Scaling Advantage

- Demonstration of Algorithmic Quantum Speedup for an Abelian Hidden Subgroup Problem

Ising Model

- Mathematical model for a ferromagnetism in statistical mechanics.
- The energy of spin configuration for a given lattice is given by the following classical Hamiltonian

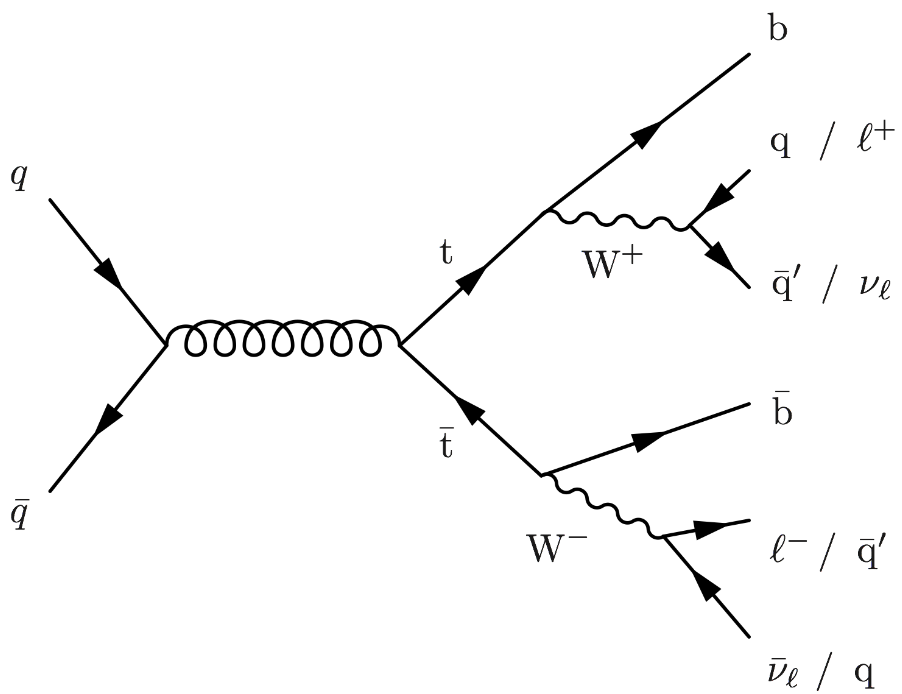
$$E(s) = -\sum_{i,j} J_{ij}\, s_i\, s_j - \sum_i h_i\, s_i \quad s = \{s_i\}, \quad s_i \in \{-1,1\}$$

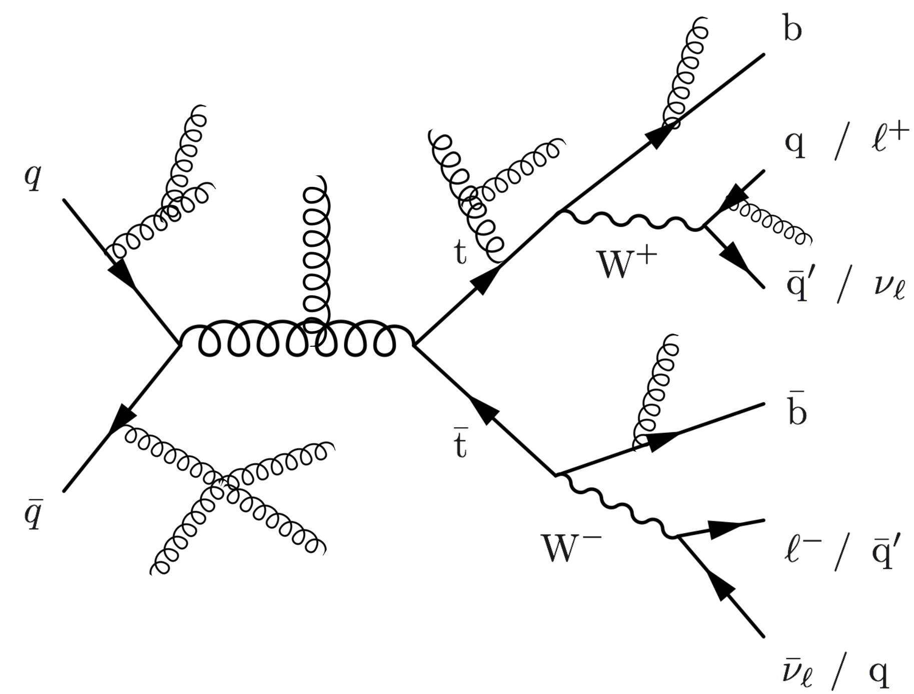
- J_{ij} is called an interaction, spin-spin coupling, and h_i is an external magnetic field, interacting with spin s_i.
- The configuration probability is given by the Boltzmann distribution

$$P(s) = \frac{e^{-\beta H(s)}}{\sum_s e^{-\beta H(s)}}, \qquad \beta = \frac{1}{k_B T}$$

- Quantum Ising model: $H = -\sum_{i,j} J_{ij}\, \sigma_i^z\, \sigma_j^z - \sum_i h_i\, \sigma_i^z$

Combinatorial problems in the top quark production





- b-ℓ ambiguity
  (Semi-leptonic and dilepton)

Ht, mbl	CDF
pt, mbl	1009.2751, Rajaraman, Yu
MT2, mbl	1109.1563, Baringer, Kong, McCaskey, Noonan
MT2, mbl, MAOS, hybrid	1109.2201, Kim, Guadagnoli, Park
M2, mbl, hybrid	1706.04995, Debnath, Kim, Kim, Kong, Matchev
NN	2202.05849, Alhazmi, Dong, Huang, Kim, Kong, Shih

- Fully hadronic channel: $2^6 = 64$ possibilities for 6 particles in the final state. But in reality, 10-20 jets appear in the final state, leading to 2^{10}-2^{20} possibilities.