LA-UR-21-31310

# *A brief introduction to quantum machine learning.*

Marco Cerezo

Information Sciences CCS-3

@Physics Without Frontiers: Quantum Machine Learning

cerezo@lanl.gov

Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

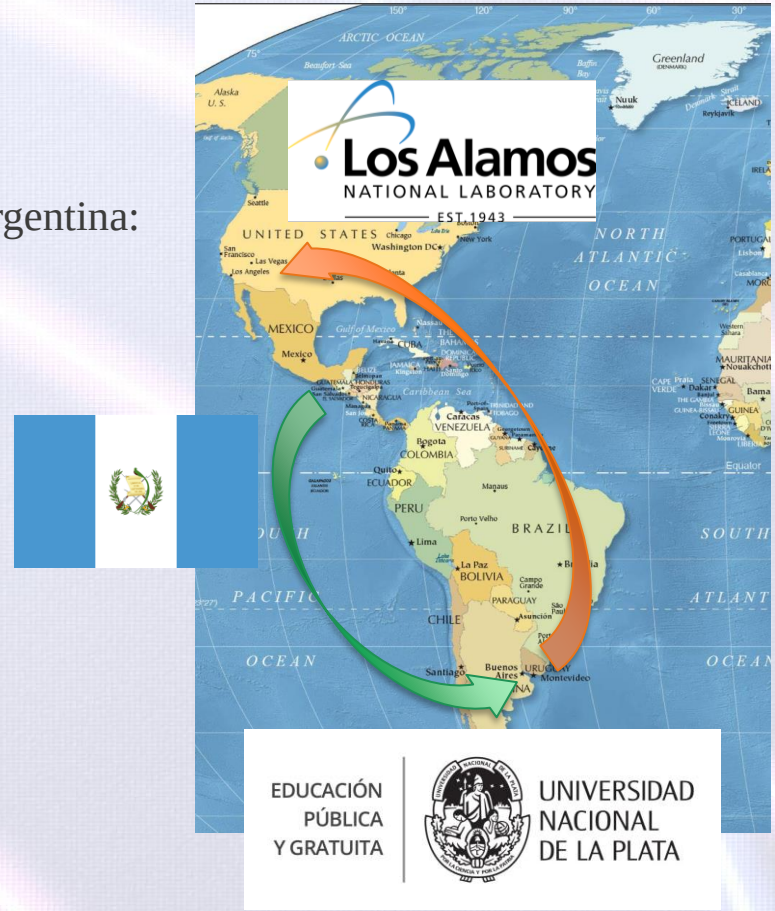
# First is first: context about myself.

From Guatemala.

Undergrad and PhD @Universidad National de la Plata, Argentina:
*Condensed matter, Quantum information, Foundations of quantum mechanics.*

Postdoc at Los Alamos National Laboratory (LANL):
*Near-term quantum computing*

Staff Scientist Information Sciences (LANL):
*Quantum machine learning.*



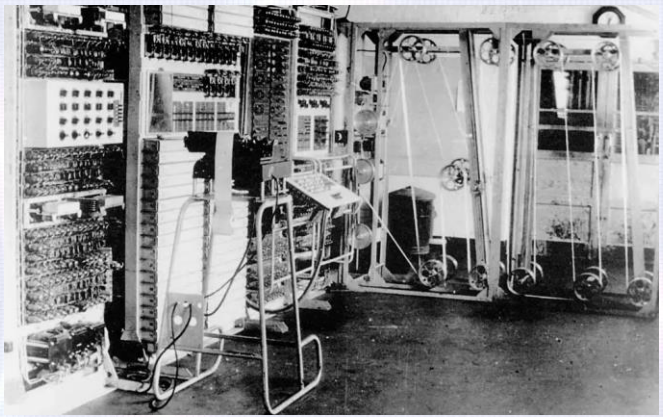
# Outline

- Why Quantum Computing?
- The promise of Quantum Computing
- Quantum Computing in the near-term.
- Quantum Machine Learning

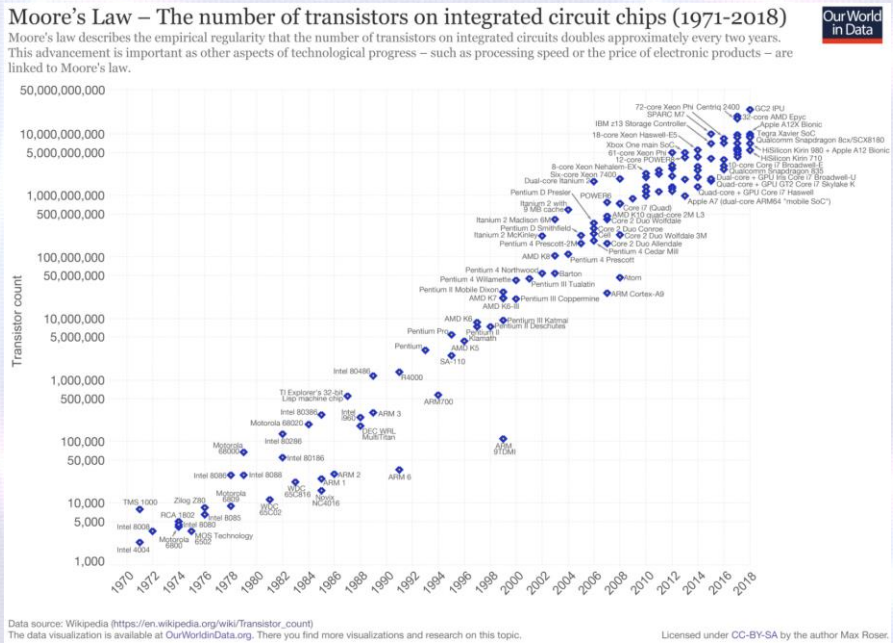
# Classical Computers

The idea of a programmable computer dates to the 19th century, with Charles Babbages studies on mechanical computers.

The evolution of the modern computer has involved a series of changes from one type of physical realization to another: from gears to relays to transistors to integrated circuits…

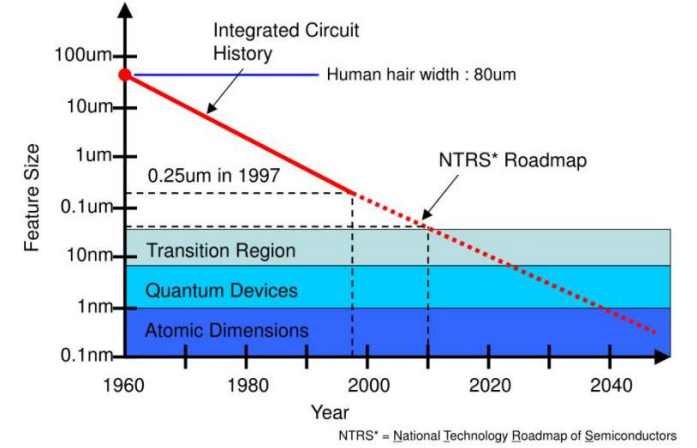


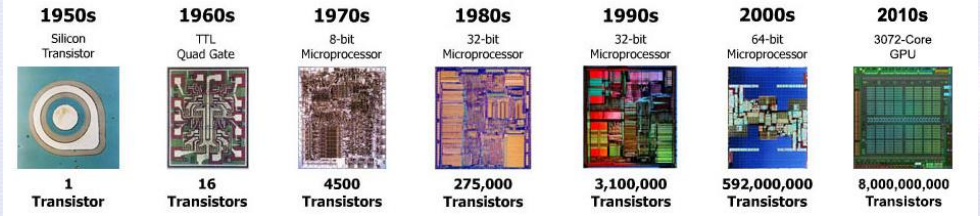
1943: ***The Colossus***, world's first programmable, electronic, digital computer, first electromechanical computer



# The end of Moore's Law?

Earlier in 2019, Nvidia CEO Jensen Huang declared that Moore's Law is no longer possible*.




Making transistors smaller runs into the regime where the laws of Quantum Mechanics kick in.

Detrimental for information processing!

Instead of making them smaller, run them in parallel.

How to go beyond this? Make new chips that somehow circumvent quantum mechanical effects?

Or, use the new set of rules of quantum mechanics for information processing tasks.

* https://www.cnet.com/tech/computing/moores-law-is-dead-nvidias-ceo-jensen-huang-says-at-ces-2019/

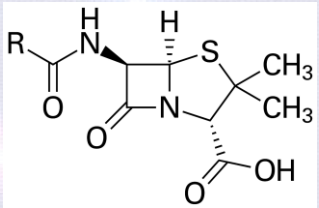
# Why do we want to use quantum mechanics for information processing?

There are certain tasks in scientifical research that require extremely large processing powers.

Studying the very phenomenon that hinders smaller transistors [Quantum Mechanical Systems] is a very expensive computational task.

$$i\hbar\frac{\partial|\psi\rangle}{\partial t} = H|\psi\rangle$$

The Hamiltonian contains information about the interactions (kinetic and potential energy): *electron-electron, nuclei-electron, nuclei-nuclei*




Modeling the structure of a molecule of an everyday drug such as the penicillin molecule:

Carbon: 4 orbitals, 16 atoms
Hydrogen: 1 orbital, 18 atoms
Nitrogen: 5 orbitals, 2 atoms = 121 orbitals …. But spin up/down = 242 spin-orbitals
Oxygen: 5 orbitals, 4 atoms
Sulfur: 9 orbitals, 1 atom

Size of $|\psi\rangle$ is $2^n = 2^{242} \sim 10^{75}$ bits of information (order of atoms in the observable universe)

# Basic idea of quantum computers

1980 when physicist Paul Benioff proposed a quantum mechanical model of the Turing machine.

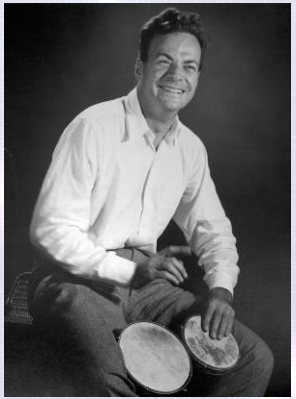
1980's Richard Feynman and Yuri Manin: use a quantum computer so simulate quantum systems.

*"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."*

It's a simple and beautiful idea.

Build a quantum computer and simulate quantum systems in it. E.g., 242 qubits could simulate the penicillin molecule as the dimension of the Hilbert space is $2^n = 2^{242}$.

But…. It's a significant technological challenge. Previous work on quantum systems required bulk control. Now we need complete control over single quantum systems.



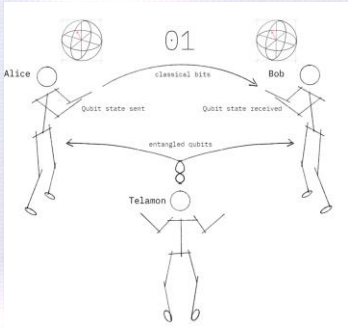
Nobel Laureate
Richard Feynman

# What *else* can we do with quantum computers?

We have a intuition that we can *simulate quantum systems* more efficiently in a quantum computer. But what about other forms of information processing tasks?

Quantum communications and information transmission:

- Quantum teleportation algorithm (teleport a quantum state).
- Superdense coding (a single qubit encodes two bits of information).
- Encryption of data based on quantum mechanics.




More importantly for us:

***Quantum algorithms***

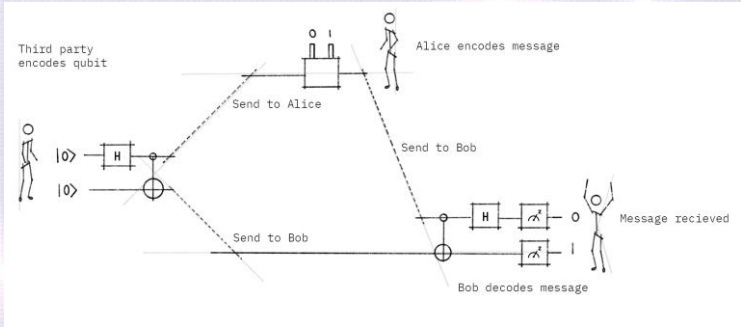



Image credit: https://qiskit.org/textbook

# Algorithmic Complexity 101

Algorithm: set of rules or steps to be followed in problem-solving operations. Can be either *classical* or *quantum*.

*Q: Given a task, can we come up with an quantum algorithm that will outperform its classical counterpart?*

First, we need to know what *outperforming* means.

Every algorithm has a *computational complexity* that is associated with either the run-time or the number of operations it needs to run.

$$C = A \cdot B$$

Where $A$, $B$ and $C$ are $d \times d$ matrices. We want to fine the elements of $C$ as $c_{ij} = \sum_{k=1}^{d} a_{ik} b_{kj}$

*Input: matrices A and B. Let C be a new $d \times d$ matrix*
*For i from 1 to d:*
    *For **j** from 1 to d:*
        *Let sum = 0*
        *For k from 1 to d:*
            *Set sum ← sum + $a_{ij} \times b_{ij}$*
        *Set $c_{ij}$ ← sum*
*Return C*

Has a computational complexity of $d^3$.

If we want to compute $\langle\psi|H|\psi\rangle$

$\langle\psi|$ is a $1 \times 2^n$ matrix
$H$ is a $2^n \times 2^n$ matrix
$|\psi\rangle$ is a $2^n \times 1$ matrix

*Computational complexity of $2^{n+1}$.*

**Generally, classical algorithms for quantum systems scale exponentially with the number of qubits n!**

# Algorithmic Complexity 101, Part 2

Classical algorithms for certain tasks have a computational complexity that *scales with the problem size*.

*We usually denote this is as* $\mathcal{O}(poly(d))$ .

We say that $f(x) \in \mathcal{O}(g(x))$ if there exists a $c > 0$ and an $x_0$ such that $f(x) \leq c\, g(x)$ for all $x \geq x_0$ .

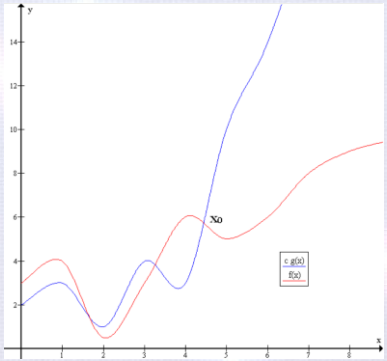



Image credit: wikipedia

As we previously saw, this is the case for classical algorithms for quantum systems of $n$-qubits scale as $\mathcal{O}(poly(d)) = \mathcal{O}(poly(2^n))$.

*Q: Given a task, can we come up with an quantum algorithm that will outperform its classical counterpart?*

Should be rephrased as

*Q: Given a task, can we come up with an quantum algorithm with a more efficient scaling than classical counterpart?*

Best case scenario $\mathcal{O}(log(d))$ **[exponential speedup], *or*** $\mathcal{O}(\sqrt[k]{d})$ **[polynomial speedup]**

# Quantum algorithms

Algorithm for *quantum simulation* can be *exponentially* faster than classical algorithm.

Surprisingly, quantum algorithm can also provide a speed-up for classical tasks!

- Finding prime factors of an integer $n$. Classical algorithm [*number field sieve*] with complexity $\mathcal{O}\left(e^{n^{1/3}\log^{2/3}(n)}\right)$.

➢ (1994) *Shor's algorithm*. Quantum algorithm $\mathcal{O}(n^2\log(n)\log(\log(n))$. This is an **exponential speed-up**!

Very important for cryptography

- Searching an untructured database of $N$ elements. Classical algorithm [*look at every element*] with complexity of $\mathcal{O}(N)$ queries

➢ (1995) *Gover's algorithm*. Quantum algorithm with $\mathcal{O}(\sqrt{N})$ queries. **Polynomial speed-up**.

Quantum computers are more powerful than Turing machines, even probabilistic Turing machines.

# More quantum algorithms

Quantum algorithms with provable spee-ups have been developed for other tasks such as:

- Fourier transform [D. Coppersmith 1994]
- Solving linear systems of equations [AW Harrow 2009]
- Principal component analysis [S Lloyd 2014]
- Semi-definite programming [FGSL Brandao 2016]
- And many more in https://quantumalgorithmzoo.org/

In general, creating quantum algorithms is *hard* due to the *counter-intuitive nature* of quantum mechanics.

We have the *theory*….

but we still need the *experimental device*.

# What do we need for a quantum computer?

According to *DiVincenzo's criteria*, constructing a quantum computer requires that the experimental setup meet five conditions necessary for quantum computation:

1. A scalable physical system with well characterized *qubit*.
2. The ability to *initialize* the state of the qubits to a simple fiducial state.
3. Long relevant *decoherence* times.
4. A "universal" set of *quantum gates*.
5. A qubit-specific *measurement* capability.

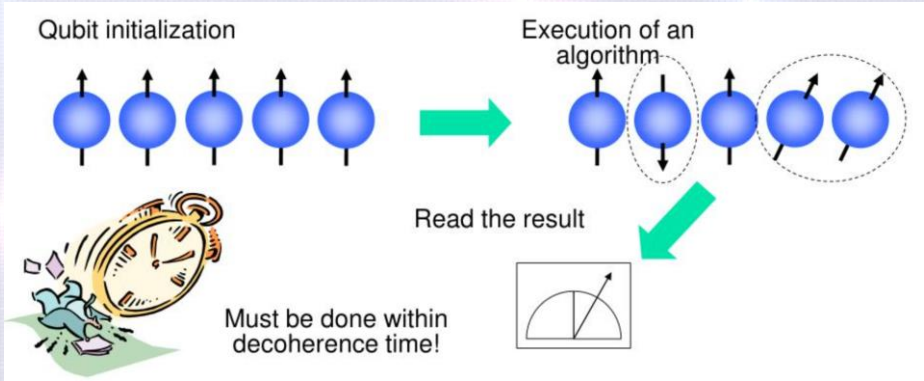



Image credit: Rod Van Meter

# So… what *is* a Qubit?

In classical computing, the fundamental unit of information is the *bit*. A bit can take two values, “0” and “1”. A bit can have many physical realizations [magnetic domain, pits in a CD, etc…]

In quantum computing, the *qubit* is the fundamental unit of information. A qubit can also take two value, “0” and “1” but it can also be in a *complex linear superposition*.

Qubit = mathematical concept, a normalized to 1 vector in a two-dimensional complex linear vector (Hilbert) space.

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Such that $|a|^2 + |b|^2 = 1$ ($|a|^2$ is the probability of qubit in $|0\rangle$).

Quantum mechanics is a *generalization of probability theory.*

A qubit can have many *physical realizations*:

- Polarization of a photon.
- Spin of an electron or a nucleus.
- Localization of a charge.
- Any two-level quantum systems.

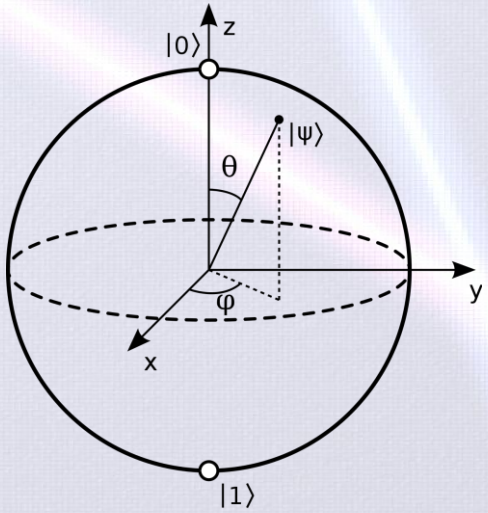
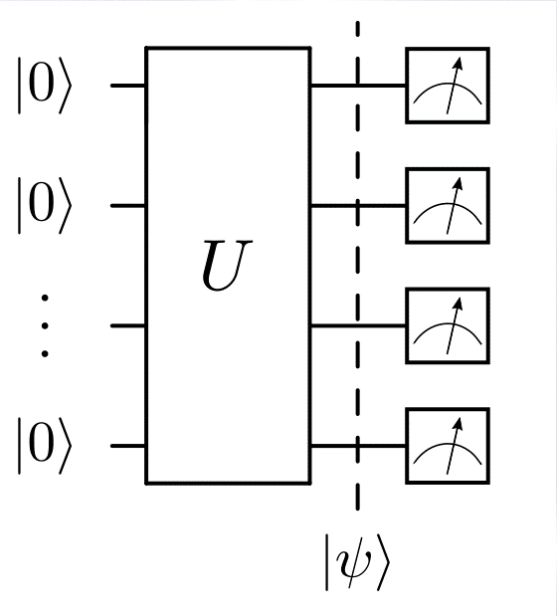



Image credit: wikipedia

# Why do qubits allow for a computational advantage?

The state of 1 qubit can be in a superposition of 2 states in the basis $\{|0\rangle, |1\rangle\}$.
Given 2 qubits, their state can be in a superposition of 4 states $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$.
Given $n$ qubits, their state can be in a superposition of $2^n$ states $\{|0\rangle, |1\rangle\}^{\otimes n} = \{|0 \dots 0\rangle, \dots, |1 \dots 1\rangle\}$.

- The dimension of the Hilbert space grows *exponentially* with the number of qubits.
- *Quantum gates* between qubits perform mathematical operations on binary data.
- Quantum states are in a large superposition that contain *entanglement*, i.e., quantum correlations between binary data.
- Quantum *interference* amplifies probability of desired outputs (answers).



# How can we realize a Qubit?

The idea of quantum computers pushes the very limits of our technological capacities.

Prior to the 1970s typically involved a gross level of control over a *bulk* sample containing an enormous number of quantum mechanical systems, none of them directly accessible.

But we care about controlling *individual* quantum systems. Since the 1970s, there have been tremendous developments in state-of-the-art technologies for the control of quantum systems.

Development of theory and experiment of quantum computing: ***Second Quantum Revolution***

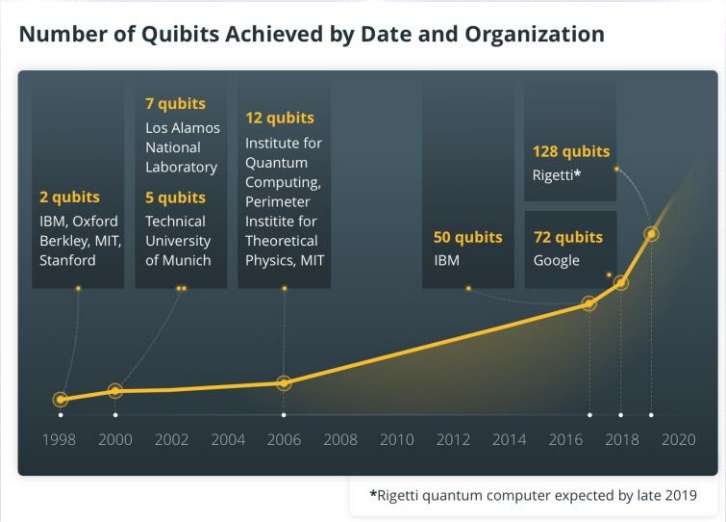
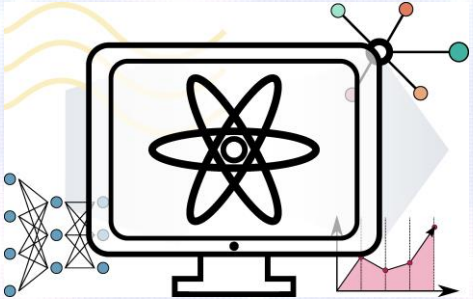


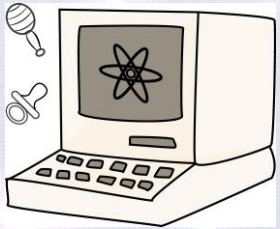

Image credit: cointelegraph.com

# The NISQ Era.


We already know that certain task are beyond classical computers.



And that quantum computers can bring an advantage.



Current quantum devices are still in their infancy,

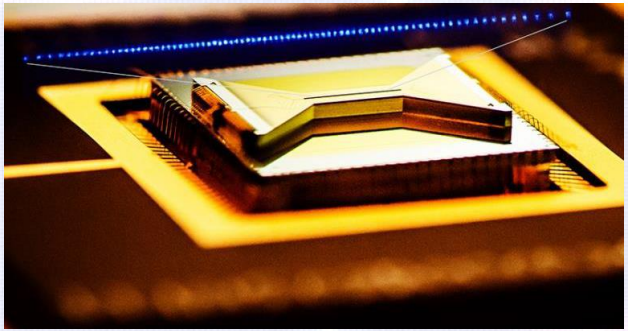
We are in the so-called ***Noisy Intermediate-Scale Quantum (NISQ)*** era.

Quantum devices are:

- *Small number of qubits* (tens of qubits).
- *Error prone* (qubit-environment interaction). → *Limited depth*.

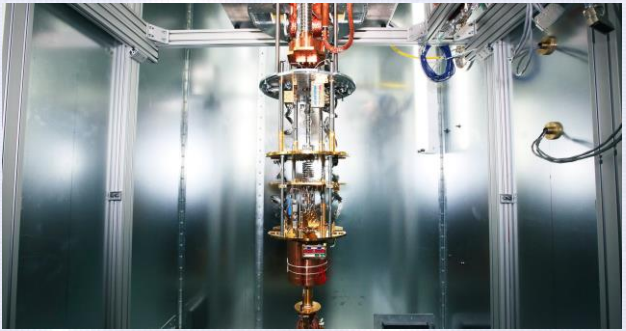
# Currently available quantum computers.

Photonic Quantum Computer
Photo: Xanadu – 8 qubits – Cloud access



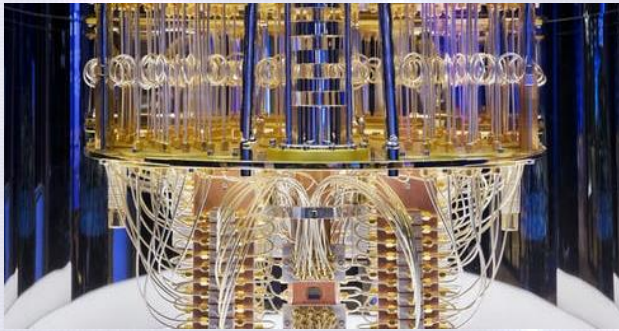
Trapped Ions Quantum Computing
Photo: IonQ - 32 qubits


IBM Superconducting qubits
Photo: IBM – 65 qubits – Cloud access



Google Superconducting qubits
Photo: Google – 72 qubits

Zuchongzhi Superconducting qubits
Photo: UST of China – 66 qubits



Rigetti Superconducting qubits
Photo: Rigetti – 32 qubits – Cloud access

# “Textbook” algorithms in the NISQ era.

Can we use currently-available NISQ devices for Fourier transform, or for breaking the RSA encryption system?

PRL 103, 150502 (2009) PHYSICAL REVIEW LETTERS week ending 9 OCTOBER 20(

**Quantum Algorithm for Linear Systems of Equations**

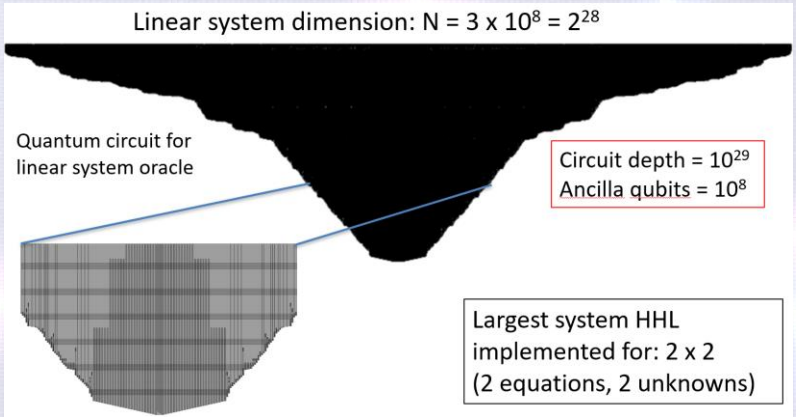
Aram W. Harrow,[1] Avinatan Hassidim,[2] and Seth Lloyd[3]
[1]*Department of Mathematics, University of Bristol, Bristol, BS8 1TW, United Kingdom*
[2]*Research Laboratory for Electronics, MIT, Cambridge, Massachusetts 02139, USA*
[3]*Research Laboratory for Electronics and Department of Mechanical Engineering, 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

**Concrete resource analysis of the quantum linear system algorithm used to compute the electromagnetic scattering cross section of a 2D target**

Artur Scherer*[1], Benoît Valiron[2], Siun-Chuon Mau[1], Scott Alexander[1], Eric van den Berg[1], and Thomas E. Chapuran[1]
[1]*Applied Communication Sciences, 150 Mt Airy Rd., Basking Ridge, NJ 07920*
[2]*CIS Dept, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104-6389**

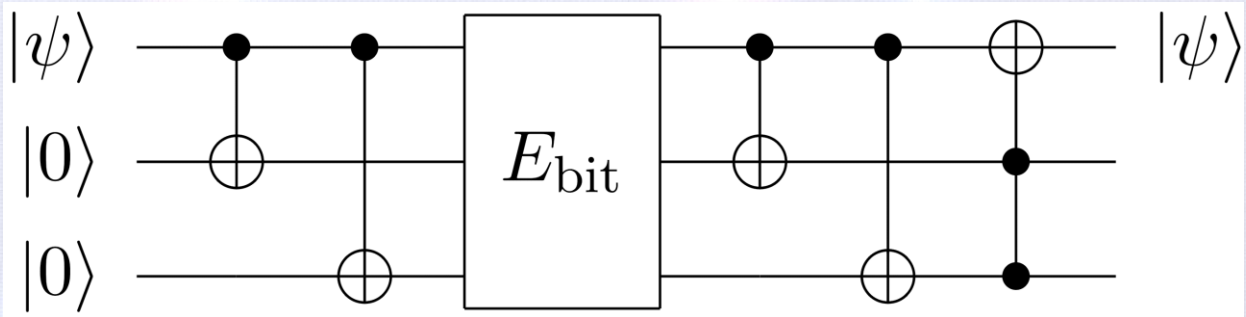
We provide a detailed estimate for the logical resource requirements of the quantum linear system algorithm [Harrow et al., Phys. Rev. Lett. **103**, 150502 (2009)] including the recently described elaborations and application to computing the electromagnetic scattering cross section of a metallic target [Clader et al., Phys. Rev. Lett. **110**, 250504 (2013)]. Our resource estimates are based on the



# What about quantum error correction?

*Quantum error correction (QEC)* is used in quantum computing to protect quantum information from errors due to decoherence and other quantum noise. It allows us to *correct an arbitrary single-qubit quantum error*.

Similarly to error correction in classical computing, QEC creates *redundancies*, and leveraging *entanglement* to detect and *correct syndromes*.




How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney[1] and Martin Ekerå[2]

[1]Google Inc., Santa Barbara, California 93117, USA
[2]KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
Swedish NCSA, Swedish Armed Forces, SE-107 85 Stockholm, Sweden

We significantly reduce the cost of factoring integers and computing discrete logarithms in finite fields on a quantum computer by combining techniques from Shor 1994, Griffiths-Niu 1996, Zalka 2006, Fowler 2012, Ekerå-Håstad 2017, Ekerå 2017, Ekerå 2018, Gidney-Fowler 2019, Gidney 2019. We estimate the approximate cost of our construction using plausible physical assumptions for large-scale superconducting qubit platforms: a planar grid of qubits with nearest-neighbor connectivity, a characteristic physical gate error rate of $10^{-3}$, a surface code cycle time of 1 microsecond, and a reaction time of 10 microseconds. We account for factors that are normally ignored such as noise, the need to make repeated attempts, and the spacetime layout of the computation. When factoring 2048 bit RSA integers, our construction's spacetime volume is a hundredfold less than comparable estimates from earlier works (Van Meter et al. 2009, Jones et al. 2010, Fowler et al. 2012, Gheorghiu et al. 2019). In the abstract circuit model (which ignores overheads from distillation, routing, and error correction) our construction uses $3n + 0.002n \lg n$ logical qubits, $0.3n^3 + 0.0005n^3 \lg n$ Toffolis, and $500n^2 + n^2 \lg n$ measurement depth to factor $n$-bit RSA integers. We quantify the cryptographic implications of our work, both for RSA and for schemes based on the DLP in finite fields.

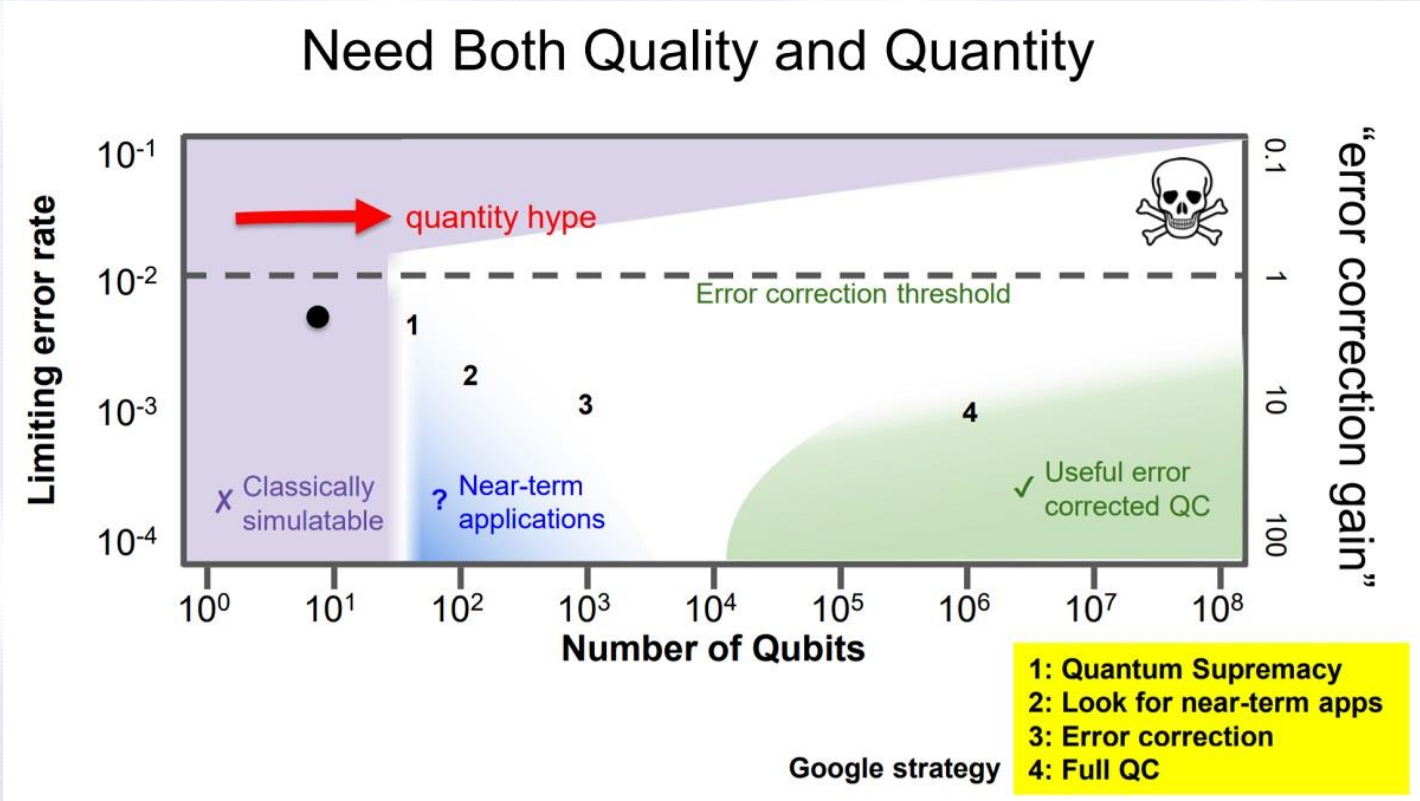
# Fault-tolerant vs NISQ

- Reliably prepare qubits and implement quantum operations.
- Scalable device, thousands, or even millions of qubits.
- Error correction to arbitrarily suppress error rates.

- Errors in qubit preparation and gate implementation.
- Tens of qubits to few hundred qubits.
- Can't fully implement error correction.

Limit the depth and size of the algorithms (gates and measurements). Can't implement many (textbook) algorithms.

# Is more better?




Number of qubits isn't the only metric!

# Can we still get something useful out of NISQ devices?

*YES!*

- **Foundational science perspective**:

Despite their limitations, current quantum devices allow us to *probe the nature of individual quantum* systems to unprecedented levels.

- **Computational perspective**:

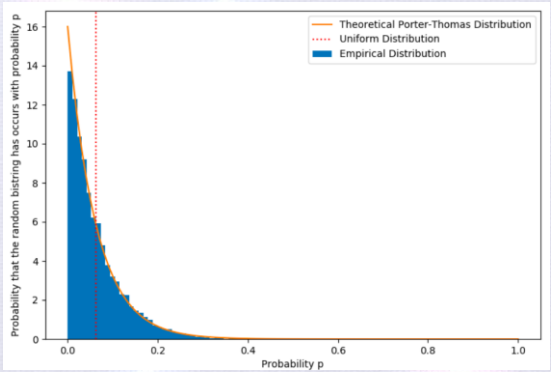
In 2019, Google achieved *Quantum Supremacy*, where a quantum computer was used to solve a contrived mathematical task faster than any classical algorithm.

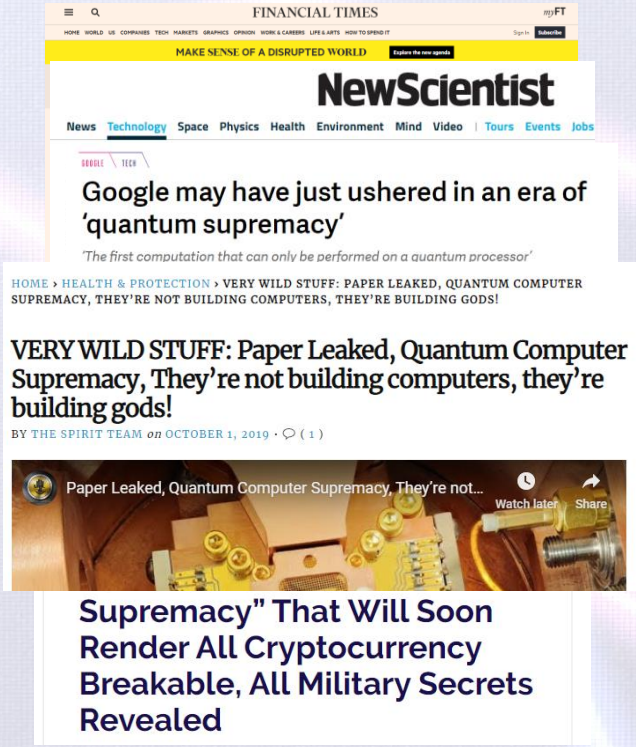
**Problem**: Generate a random quantum circuit.
Apply it to an initial state and measure some elements from the output distribution ($2^n$ possible outcomes).

Should converge to a Porter-Thomas distribution.

Are they building gods: No!

Milestone: YES!



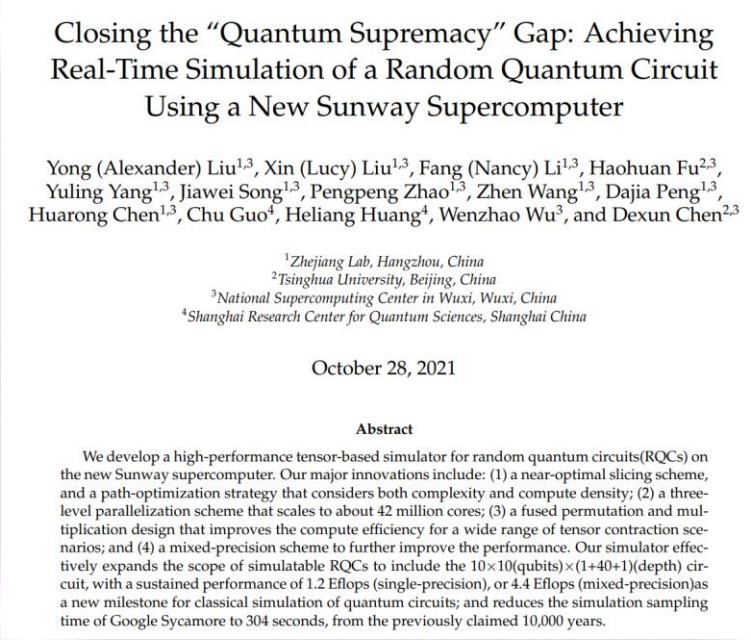
# Has the google supremacy result hold up?

*No… but that's ok.*

In their 2019, Google claims: "*Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years.*"

Newer results show: "*a new milestone for classical simulation of quantum circuits; and reduces the simulation sampling time of Google Sycamore to 304 seconds, from the previously claimed 10,000 years.*"

Classical and quantum will naturally push-back, compete and improve each other!



Closing the "Quantum Supremacy" Gap: Achieving Real-Time Simulation of a Random Quantum Circuit Using a New Sunway Supercomputer

Yong (Alexander) Liu[1,3], Xin (Lucy) Liu[1,3], Fang (Nancy) Li[1,3], Haohuan Fu[2,3], Yuling Yang[1,3], Jiawei Song[1,3], Pengpeng Zhao[1,3], Zhen Wang[1,3], Dajia Peng[1,3], Huarong Chen[1,3], Chu Guo[4], Heliang Huang[4], Wenzhao Wu[3], and Dexun Chen[2,3]

[1]*Zhejiang Lab, Hangzhou, China*
[2]*Tsinghua University, Beijing, China*
[3]*National Supercomputing Center in Wuxi, Wuxi, China*
[4]*Shanghai Research Center for Quantum Sciences, Shanghai China*

October 28, 2021

**Abstract**

We develop a high-performance tensor-based simulator for random quantum circuits(RQCs) on the new Sunway supercomputer. Our major innovations include: (1) a near-optimal slicing scheme, and a path-optimization strategy that considers both complexity and compute density; (2) a three-level parallelization scheme that scales to about 42 million cores; (3) a fused permutation and multiplication design that improves the compute efficiency for a wide range of tensor contraction scenarios; and (4) a mixed-precision scheme to further improve the performance. Our simulator effectively expands the scope of simulatable RQCs to include the 10×10(qubits)×(1+40+1)(depth) circuit, with a sustained performance of 1.2 Eflops (single-precision), or 4.4 Eflops (mixed-precision)as a new milestone for classical simulation of quantum circuits; and reduces the simulation sampling time of Google Sycamore to 304 seconds, from the previously claimed 10,000 years.

# What's after supremacy? Advantage!

*Quantum Supremacy* solved a mathematical task with no immediate application.

Now we want to use a quantum computer for a *Quantum Advantage*:

***Using a quantum computer to solving a practical task of scientific, engineering, or economical interest faster than any classical super computer.***

How will we achieve this: ***HYBRID METHODS!***

NISQ Quantum Computer

Classical Computers

*Rather than competing, collaborating*

# Near-term Quantum Computing Models

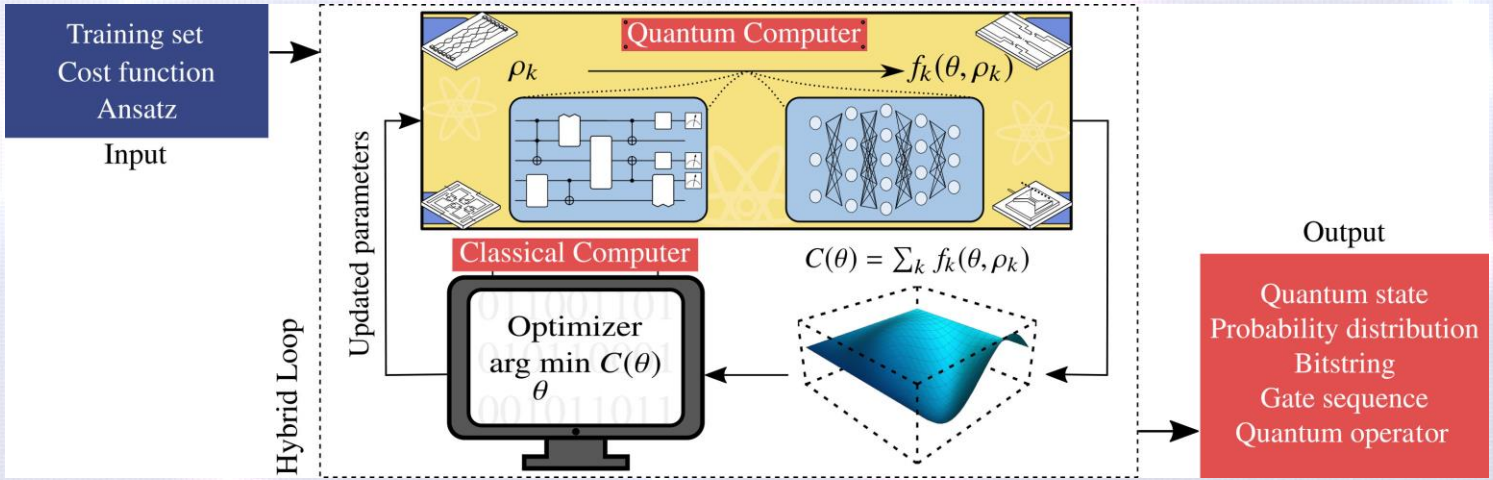
Accounting for all of the *constraints* imposed by NISQ computers with a single strategy requires an *optimization-based or learning based approach*.

Hence, ***encode*** a problem of interest into an optimization task:

Use a near-term quantum computer to ***evaluated quantities*** that a classical computer cannot efficiently evaluate.

Use ***classical optimizers*** in a classical computer to solve the optimization task.

***Analog to the highly successful machine-learning methods***!




*M. Cerezo, et al. "Variational quantum algorithms." Nature Reviews Physics (2021): 1-20.*

# Variational Quantum Algorithms (VQA)

First approach to near-term quantum computing.

Analogous to traditional programming: *Task-oriented*, user manually formulate or code rules.


# Quantum Machine Learning (QML)

Recently attracted considerable attention as a framework for near-term quantum computing.

Analogous to classical machine learning: *automated learning* process.

# VQA example: Variational Quantum Eigensolver

The most famous VQA.

***Input***: An $n$-qubit Hamiltonian describing the interactions of some physical system $H = \sum c_i h_i$

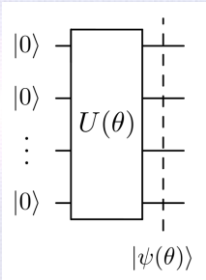
***Goal***: Find the ground-state of $H$: $H|\psi\rangle = E_{GS}|\psi\rangle$

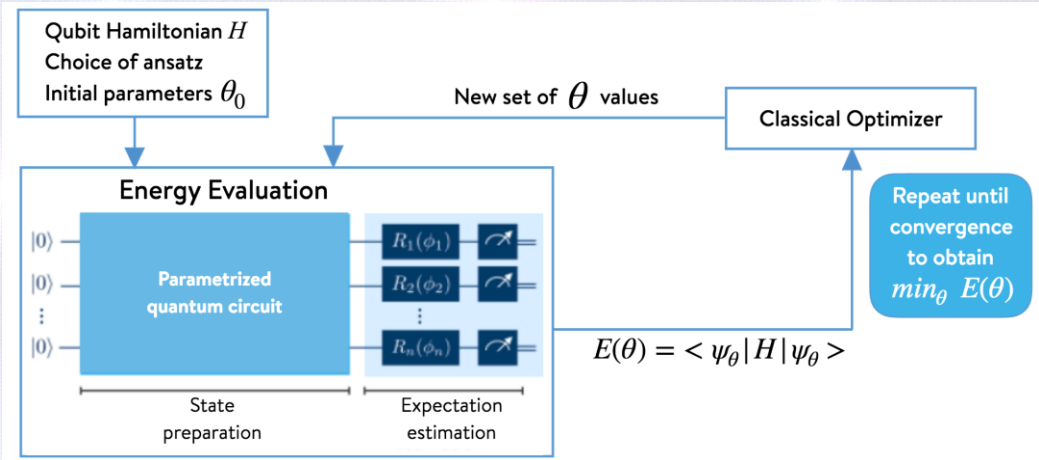
***Algorithm***:

- Initialize qubits in a simple state such as $|0\rangle^{\otimes n}$.
- Send state trough a parametrized quantum circuit $\mathrm{U}(\theta)$, whose action is given by $|\psi(\theta)\rangle = \mathrm{U}(\theta)|0\rangle^{\otimes n}$.
- Compute expectation value $\langle\psi(\theta)|H|\psi(\theta)\rangle = \sum c_i \langle\psi(\theta)|h_i|\psi(\theta)\rangle$ in a quantum computer.
- Use a classical optimizer to update $\theta$.

***Output***:

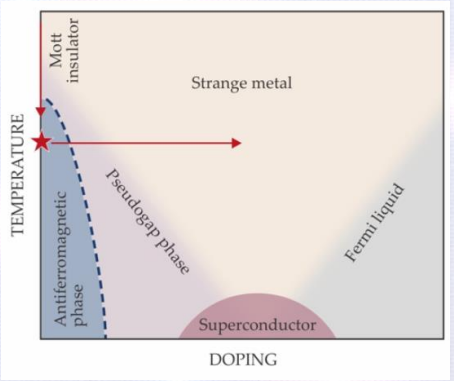
Circuit $\mathrm{U}(\theta)$ that prepares the ground-state of $H$.




*A. Peruzzo, et al. "A variational eigenvalue solver on a photonic quantum processor." Nature communications 5.1 (2014): 1-7. Image credits: openqemist*

# QML example: Quantum State Classification



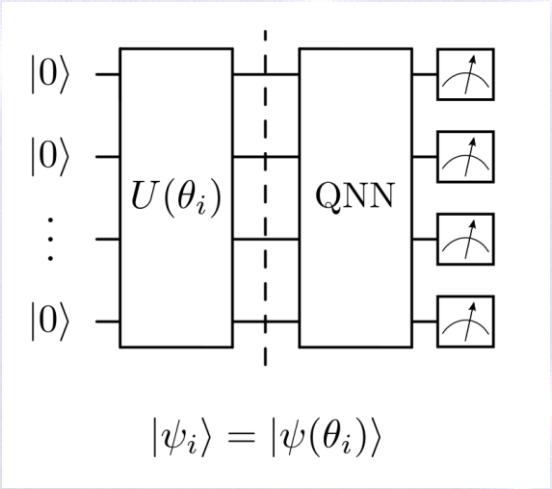

Similar to classical machine learning, we can think about a supervised learning task.

***Input***: Dataset of labeled ground states of an $n$-qubit Hamiltonian that has several quantum phases: $\{|\psi_i\rangle, y_i\}$, where $|\psi_i\rangle \in \mathcal{H}$ and $y_i \in \mathcal{Y}$ .

***Goal***: Classify states according to their phase.

Note that we can use VQE to prepare the dataset!

***Algorithm***:
Train a QML model $M: \mathcal{H} \rightarrow \mathcal{Y}$ over the dataset.
The information processing is done by a Quantum Neural Network (QNN).
For instance, use a mean-squared error or log-likelihood loss function.



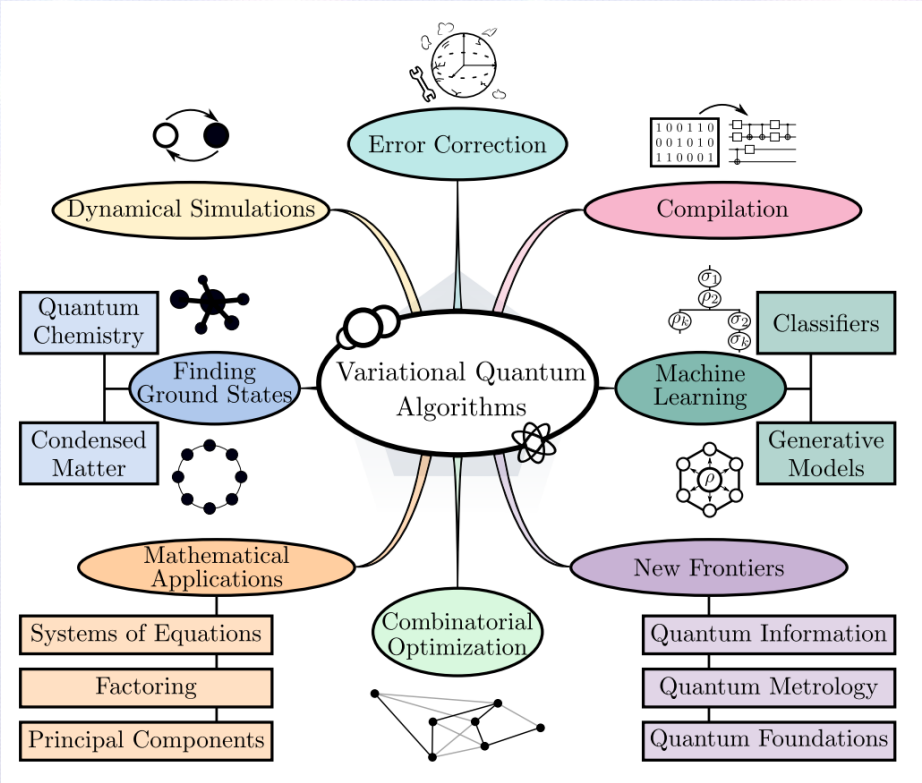

***Output***:
Quantum phase classifier.

The *learning approach is extremely natural for quantum data*.
Due to the *counter-intuitive nature* of quantum mechanics, designing quantum algorithms is a hard-task.

*Image credits: Phase diagram - A new phase for the Fermi–Hubbard model*

# More general VQA applications

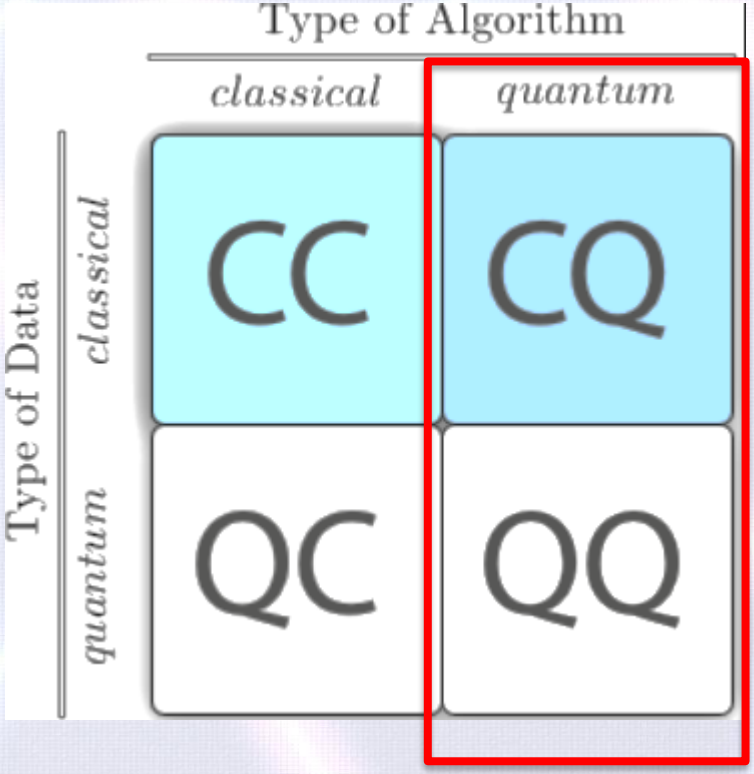
VQAs have already been considered for a plethora of applications, covering essentially ***all of the applications that researchers had envisioned for quantum computer***.




*M. Cerezo, et al. "Variational quantum algorithms." Nature Reviews Physics (2021): 1-20.*

# More general QML applications

Quantum machine learning is a very broad term.




**Kernel!**

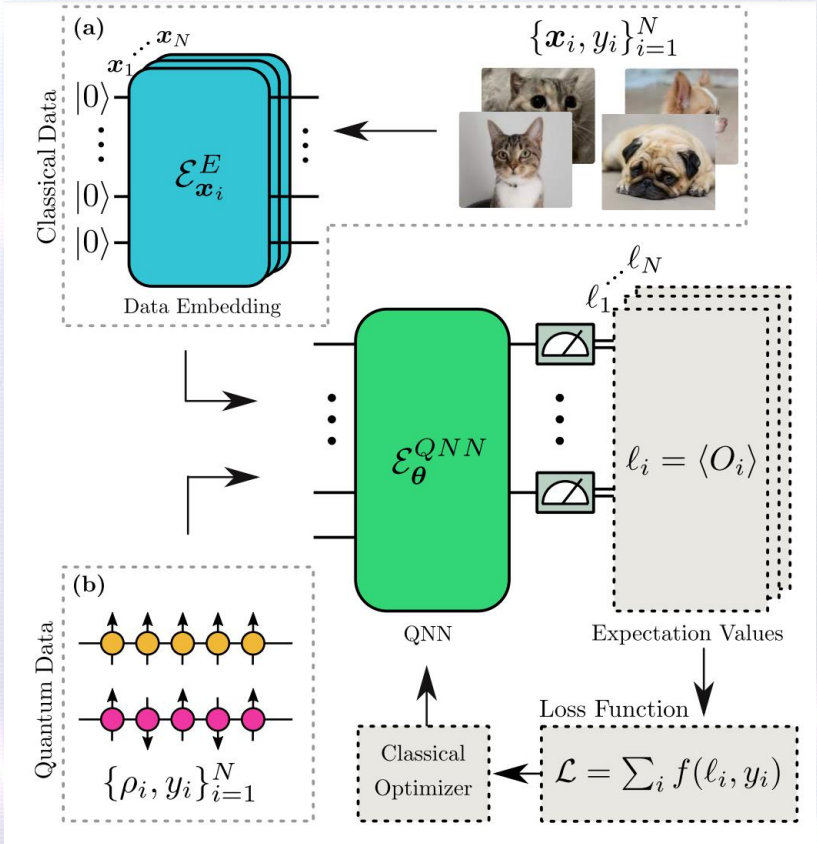
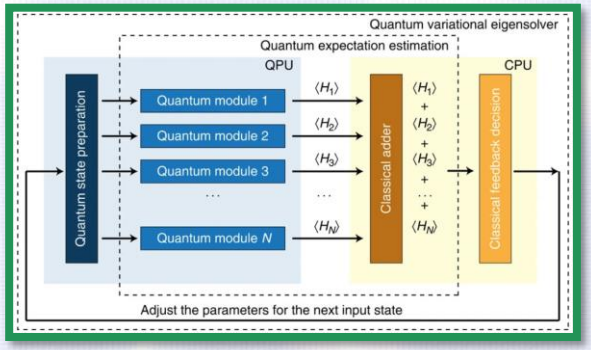


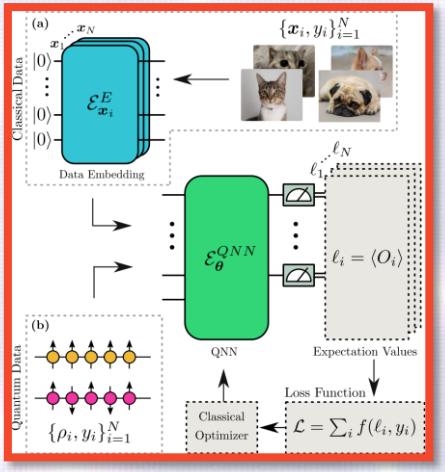

*Image credits: Wikipedia, Thanasilp, Supanut, et al. "Subtleties in the trainability of quantum machine learning models." arXiv:2110.14753 (2021).*

# Future of hybrid computers

## Variational Quantum Algorithms *vs* Quantum Machine Learning




Peruzzo, Alberto, et al. Nature communications 5.1 (2014): 1-7




**Input**: Fiduciary state, no training data.

**Task**: Minimize a cost function, linear

$$C(\boldsymbol{\theta}) = \mathrm{Tr}[U(\boldsymbol{\theta})\rho U^*(\boldsymbol{\theta})]$$

**Train**: Parametrized quantum circuit

**Input**: Non-trivial states, training data.

**Task**: Minimize a loss function, non-linear

$$L(\boldsymbol{\theta}) = \sum \mathrm{f}(\mathrm{Tr}[U(\boldsymbol{\theta})\rho_i U^*(\boldsymbol{\theta})])$$

**Train**: Quantum Neural Network

VQA

QML

?

Noisy — Partial Error Correction — Fault Tolerant

Training a QNN

# Can we guarantee a quantum speed-up with QML and VQAs?

VQAs and QML are, by definition, mostly *heuristic* methods.

Provable complexity results are case-specific and hard to come by.

Recently, there has been a great deal of effort to understand the ***capabilities and limitations*** of near-term hybrid quantum computing models.

**The power of quantum neural networks**

Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, Stefan Woerner

**Information-theoretic bounds on quantum advantage in machine learning**

Hsin-Yua

**The Inductive Bias of Quantum Kernels**

Jonas M. Kübler, Simon Buchholz, Bernhard Schölkopf

**Generalization in quantum machine learning from few training data**

Matthias C. Caro, Hsin-Yuan Huang, M. Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, Patrick J. Coles

Modern quantum machine learning (QML) methods involve variationally optimizing a parameterized quantum circuit on a training data set, and subsequently making predictions on a testing data set (i.e., generalizing). In this work, we provide a comprehensive study of generalization performance in QML after training on a limited number $N$ of training data points. We show that the generalization error of a quantum machine learning model with $T$ trainable gates scales at worst as $\sqrt{T/N}$. When only $K \ll T$ gates have undergone substantial change in the optimization process, we prove that the generalization error improves to $\sqrt{K/N}$. Our results imply that the compiling of unitaries into a polynomial number of native gates, a crucial application for the quantum computing industry that typically uses exponential-size training data, can be sped up significantly. We also show that classification of quantum states across a phase transition with a quantum convolutional neural network requires only a very small training data set. Other potential applications include learning quantum error correcting codes or quantum dynamical simulation. Our work injects new hope into the field of QML, as good generalization is guaranteed from few training data.

# Where does the field currently stand?

There is a *great deal of hope* for achieving a *quantum advantage* with VQAs and QML.

Hardware keeps improving, we can expect *better and larger devices.*

But also, there are still many unanswered questions and work that needs to be done:

- How much does *noise* in quantum-hardware limit the usability of quantum algorithms?
- How can we guarantee that we can *train* the parameters in the parametrized quantum circuit, or the quantum neural network?
- Can we guarantee that the trained QML model will *generalize* well?
- How is QML *different* than classical ML? Where can we expect a quantum advantage?
- Will QML be better with classical or quantum *data*?
- We need to develop *quantum-aware* encoding schemes, QNN architectures, optimizers, loss functions. We can't simply rely that *classically developed* methods will perform well in quantum.

*After the invention of the laser, it was called a solution in search of a problem. To some degree, the situation with QML is similar.*

**It is an exciting time, it's a young field with many unanswered questions!**

# Cool-headed approach to near-term quantum computing.

There is a lot of ***hype for quantum***. Some of which is justified… some of which maybe isn't.

We need a ***principled analysis*** so that we can avoid settling for heuristics that work on small examples, but break once larger training sets are employed.

While NN are widely used today, their historical development saw periods of great stagnation: From a single perceptron, to multi-layer perceptrons, to backpropagation methods.


Let's try to make sure we are not heading towards a quantum winter or bubble!

# What can we expect from QML and from this course?

Quantum machine learning has the potential to be a ***ground-breaking and revolutionary tool*** that can impact scientific research, industrial applications, economy, cyber-security.

Will it be available in the next couple of year?

Most like not for practical day-to-day use, but definetly yes for basic science research.

We can expect theoretical and experimental results to further our understanding of QML.

***What can we expect from this QML course?***

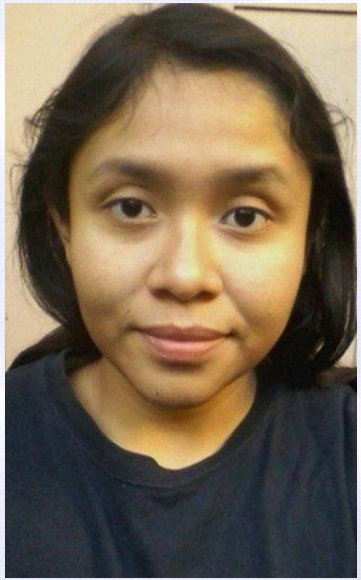
A ***theoretical and principled approach***. QML requires (at least basic) understanding of Quantum Mechanics, Quantum Information and Machine Learning.

We will cover some of the basis for all of these fields. These basic tools will be useful to understand current research, and hopefully allow you to incur and do some research in this field by yourself.

**Theoretical exercises**: basic framework for QML.

**Hands-on exercises**: coding a VQE experiment and a QML implementation for classification. Some ground-up coding!

# Teaching Assistants:



Subadra Echeverria

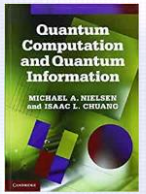
Felipe Choi

https://github.com/felipechoy1/PWF2021/

# Suggested Reading:

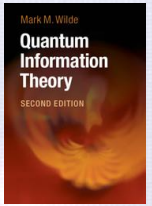
- Basic for quantum computing:

M. Nielsen, and IL. Chuang. "*Quantum computation and quantum information*.“, 2002 (online)



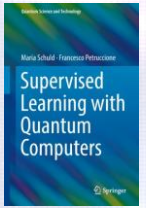

- Quantum Information:

Wilde, Mark M. “*Quantum information theory*” 2013 (https://arxiv.org/abs/1106.1445)




- Quantum Machine Learning:

M. Schuld, and F. Petruccione. “*Supervised learning with quantum computers*” 2018.




J. Biamonte, et al. "*Quantum machine learning*." Nature 549.7671 (2017): 195-202. (https://arxiv.org/abs/1611.09347)

- Variational Quantum algorithms

M. Cerezo, et al. "*Variational quantum algorithms*.“ Nature Reviews Physics (2021): 1-20. (https://arxiv.org/abs/2012.09265)

K. Bharti, "*Noisy intermediate-scale quantum (NISQ) algorithms*." 2021 (https://arxiv.org/abs/2101.08448)

S Endo, et al. "*Hybrid quantum-classical algorithms and quantum error mitigation*." Journal of the Physical Society of Japan 90.3 (2021): 032001 (https://arxiv.org/abs/2011.01382).

# Thank you for your attention!
# Next Step: pen and paper class.

cerezo@lanl.gov