Applications of quantum computing

Antonio Mezzacapo
IBM TJ Watson Research Center
Yorktown Heights, NY

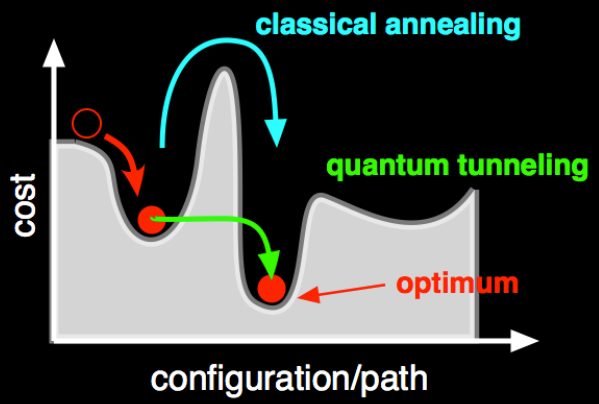
2

Types of Quantum Devices

Quantum Annealing

Optimization Problems

- Machine learning
- Fault analysis
- Resource optimization
- etc…

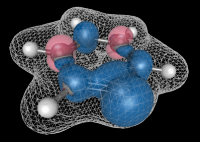


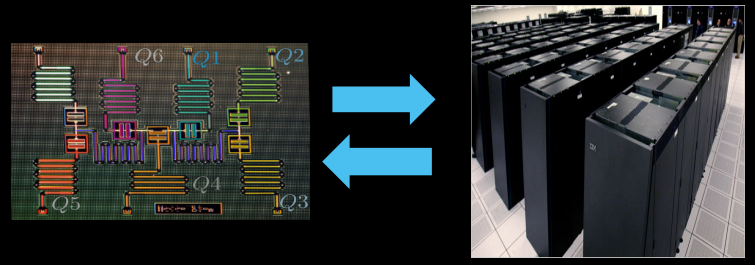
Many ‘noisy’ qubits can be built;
large problem class in optimization;
amount of quantum speedup unclear
Not a universal quantum computer

Approximate NISQ-Comp.

Simulation of Quantum Systems, Optimization

- Material discovery
- Quantum chemistry
- Optimization
  (logistics, time scheduling,…)
- Machine Learning



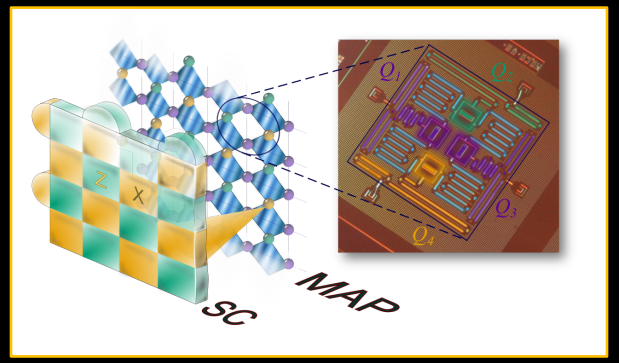


Hybrid quantum-classical approach;
already 50-100 “good” physical qubits
could provide quantum speedup.

Fault-tolerant Universal Q-Comp.

Execution of Arbitrary Quantum Algorithms

- Algebraic algorithms
  (machine learning, cryptography,…)
- Combinatorial optimization
- Digital simulation of quantum systems



Surface Code: Error correction in a Quantum Computer

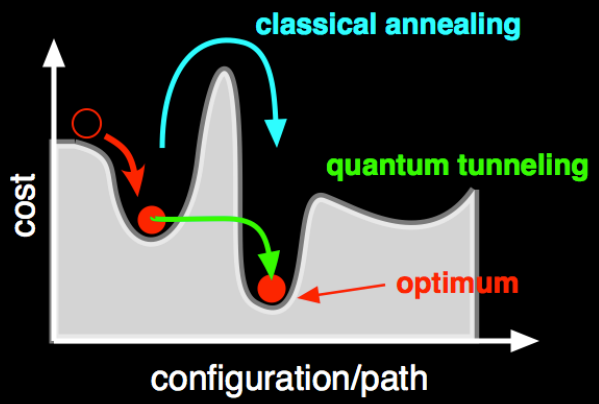
Proven quantum speedup;
error correction requires significant qubit overhead.

Types of Quantum Devices

Quantum Annealing

Optimization Problems

- Machine learning
- Fault analysis
- Resource optimization
- etc…

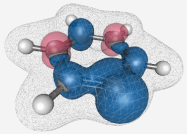


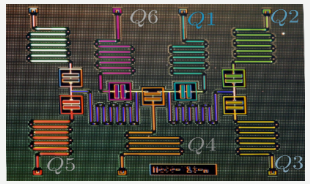
Many ‘noisy’ qubits can be built;
large problem class in optimization;
amount of quantum speedup unclear

Approximate NISQ-Comp.

Simulation of Quantum Systems, Optimization

- Material discovery
- Quantum chemistry
- Optimization
  (logistics, time scheduling,…)
- Machine Learning



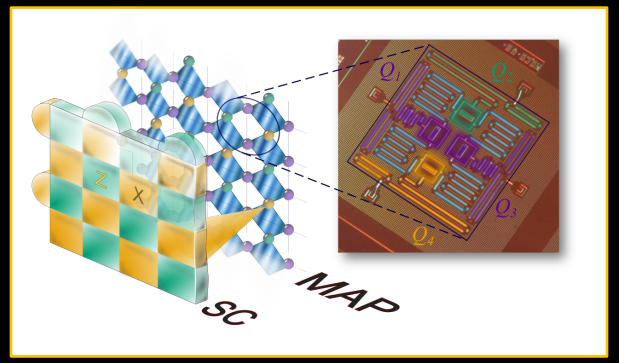


Hybrid quantum-classical approach;
already 50-100 “good” physical qubits
could provide quantum speedup.

Fault-tolerant Universal Q-Comp.

Execution of Arbitrary Quantum Algorithms

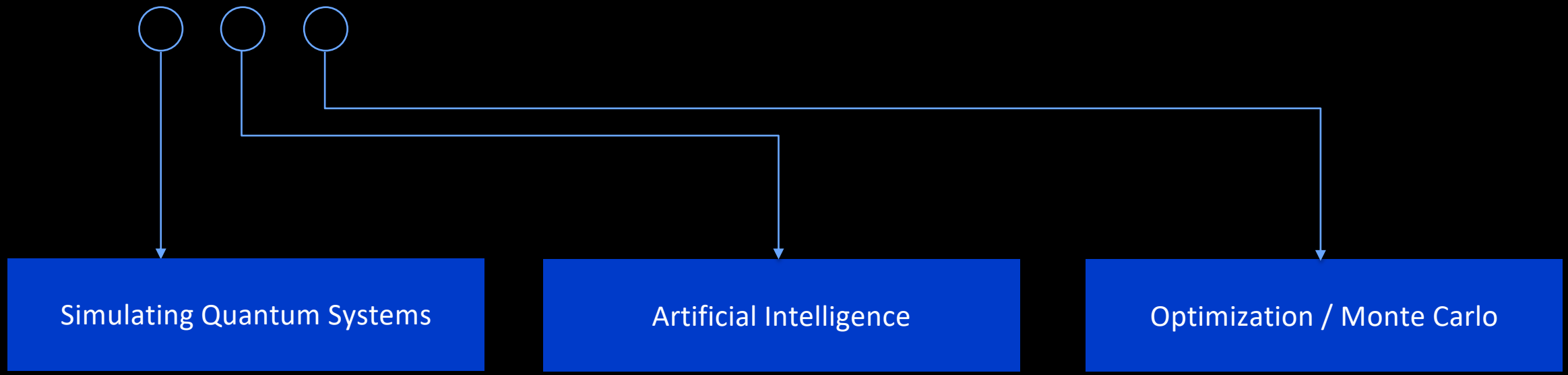
- Algebraic algorithms
  (machine learning, cryptography,…)
- Combinatorial optimization
- Digital simulation of quantum systems



Surface Code: Error correction in a Quantum Computer

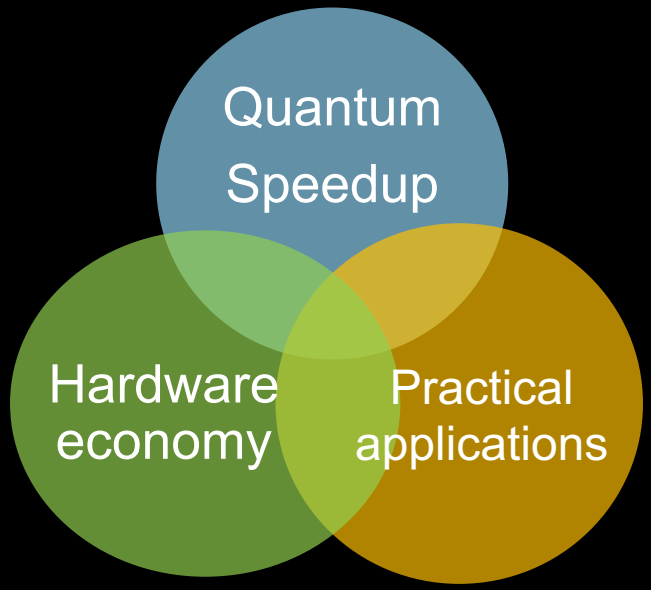
Proven quantum speedup;
error correction requires significant qubit overhead.

In collaboration with IBM Q Network partners, we are driving advancements in quantum software and algorithms.



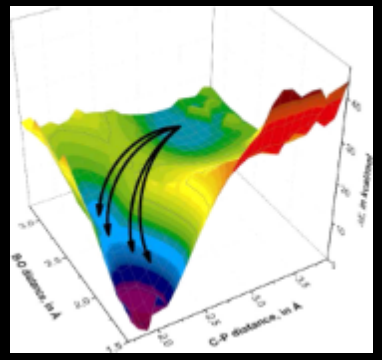
Quantum Chemistry

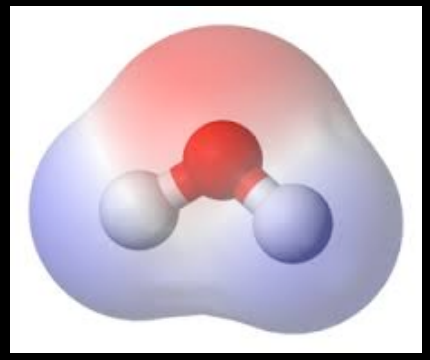
IBM Q



Quantum Chemistry: What can early quantum computers do?

Reaction rates





Molecular structure

Quantum chemistry:
Why a challenge?

Classical algorithms need resources that scale exponentially with system size for solving fermionic systems exactly.

$$H_{\mathrm{el}} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2 - \sum_{i=1}^{N_{\mathrm{el}}}\sum_{A=1}^{N_{\mathrm{nu}}}\frac{Z_A}{r_{iA}} + \sum_{i=1,j>i}^{N_{\mathrm{el}},N_{\mathrm{el}}}\frac{1}{r_{ij}}$$

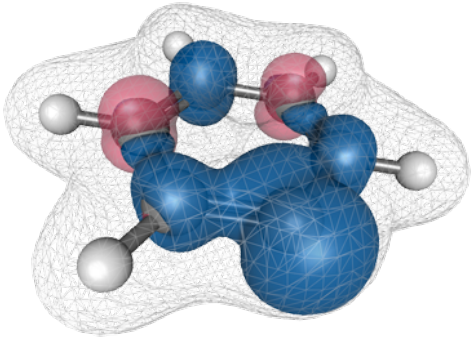
Full CI (exact): memory requirements

Classical $\mathcal{O}(\exp(N))$

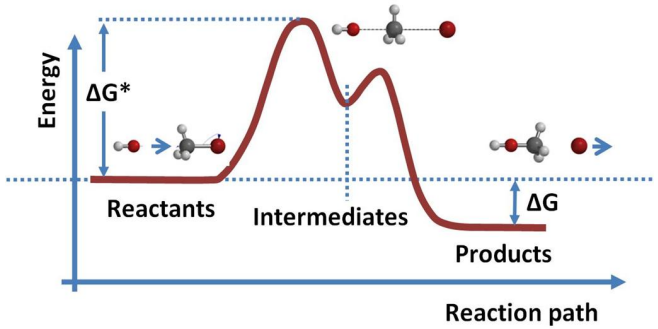
Quantum $\mathcal{O}(N)$

Sign problem: Monte-Carlo simulations of fermions are NP-hard [Troyer &Wiese, PRL 170201 (2015)]

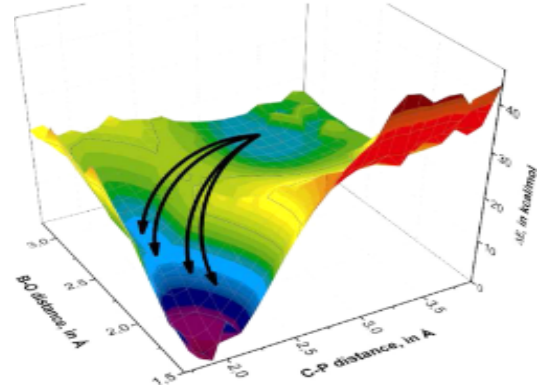
molecular structure



reaction rates



reaction pathways



Quantum – classical hybrid algorithm for near term

Problem cast into a Quantum Hamiltonian

Map the problem to qubits

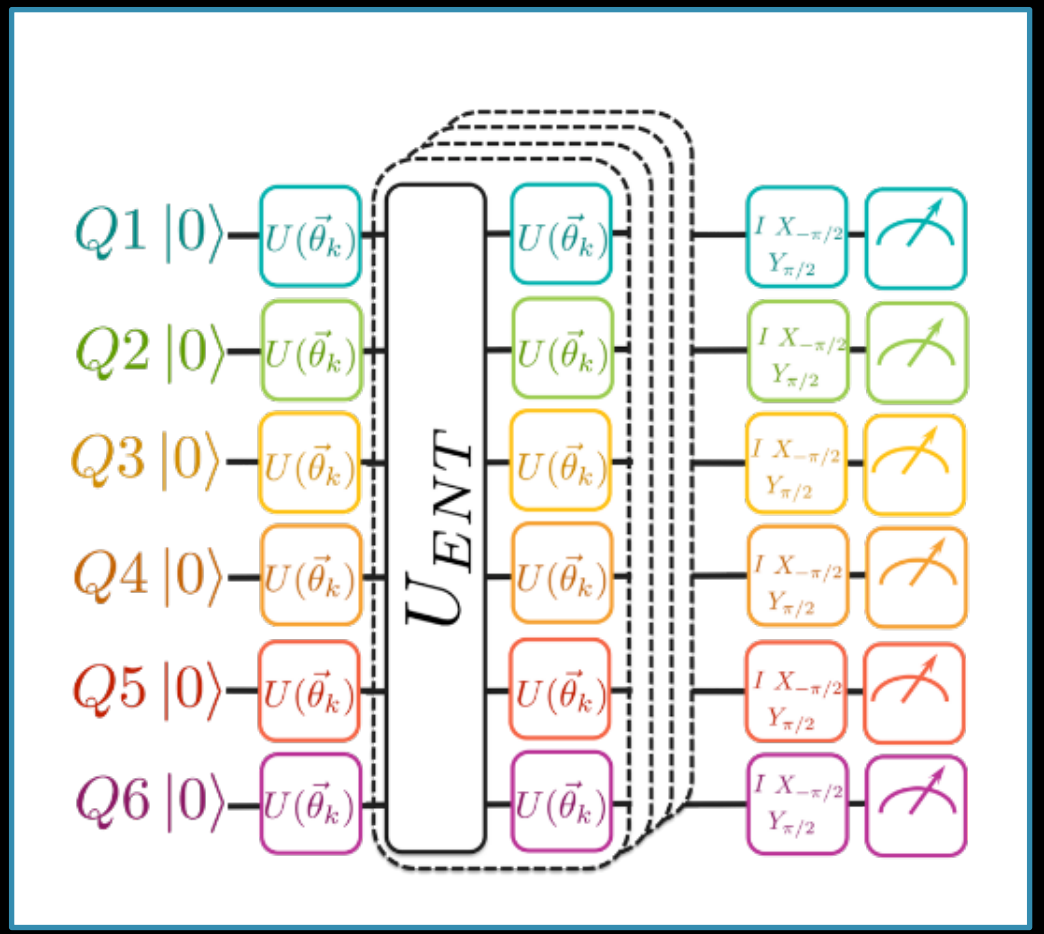
Trial State prepared on quantum chip

Measurement on quantum chip

Optimization

Solution

A quantum trial state

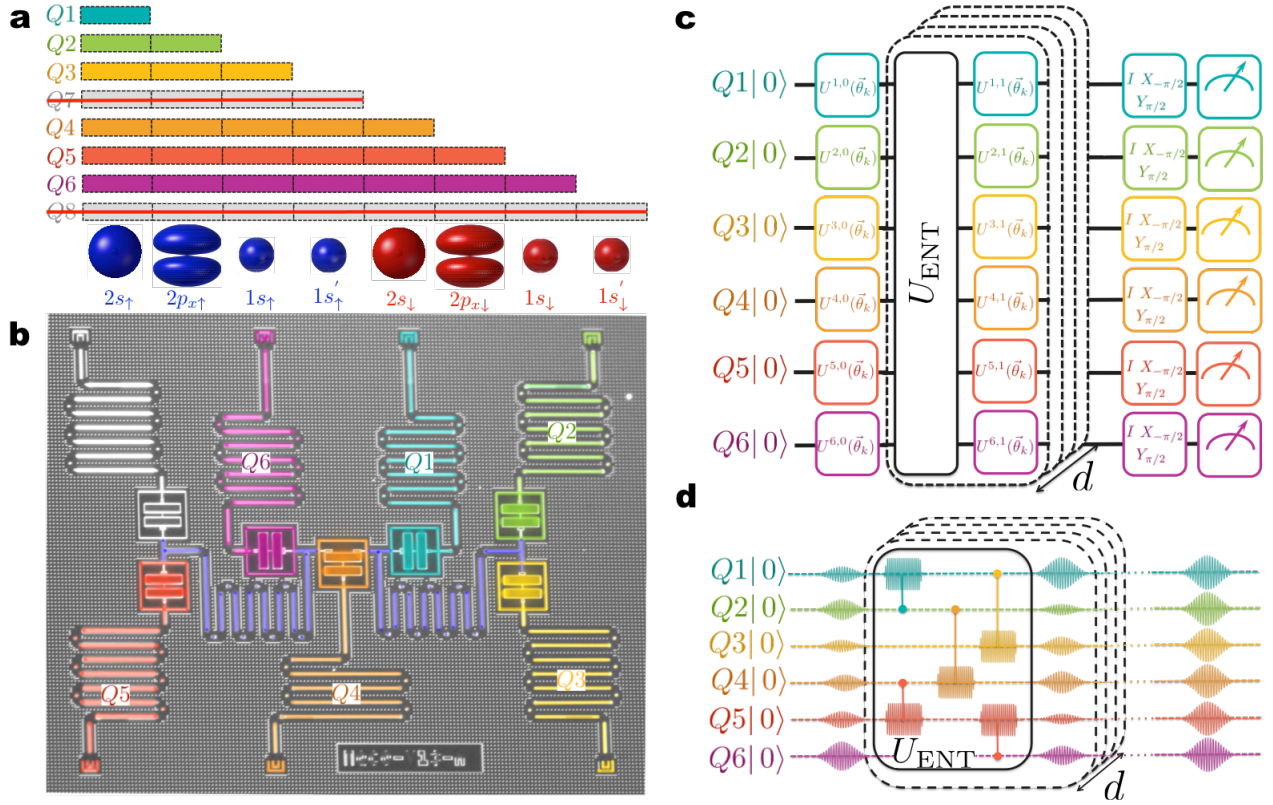


Advances in optimization of quantum chemistry problems

A. Kandala, A. Mezzacapo, K. Temme, M. Takita,
M. Brink, J. M. Chow, J. M. Gambetta
Hardware-efficient variational quantum eigensolver for
small molecules & quantum magnets
Nature **549**, 242–246 (2017)

Quantum chemistry on a superconducting quantum processor: BeH_2 with 6 qubits

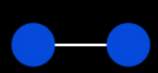
Hardware-efficient variational
trial state generation



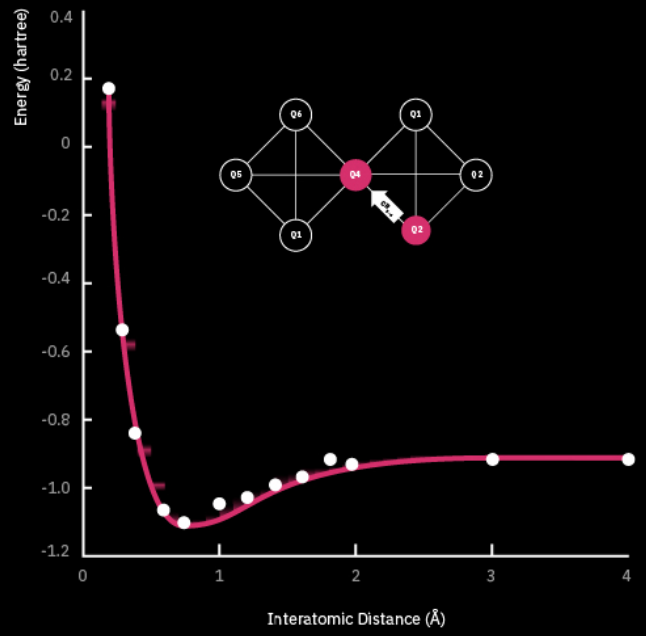
entangling gates

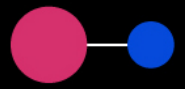
Computing the ground state energy for small molecules

A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, J. M. Gambetta. Nature **549**, 242, 2017

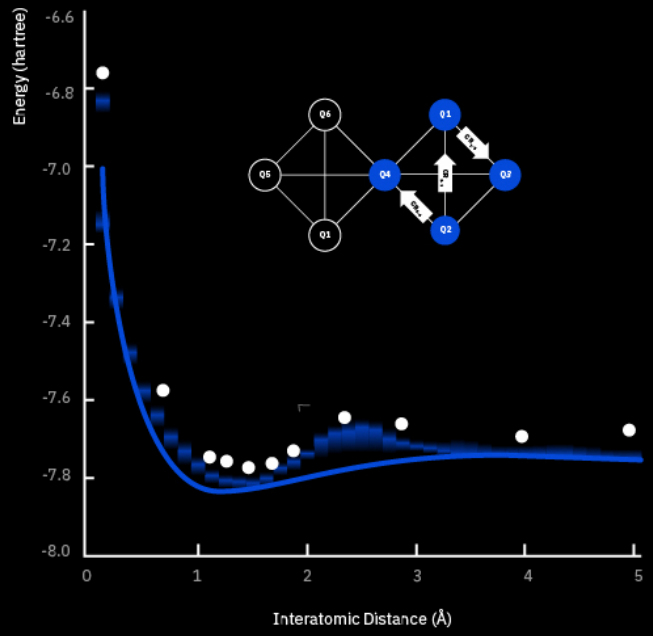


Hydrogen (H_2): 2 Qubits

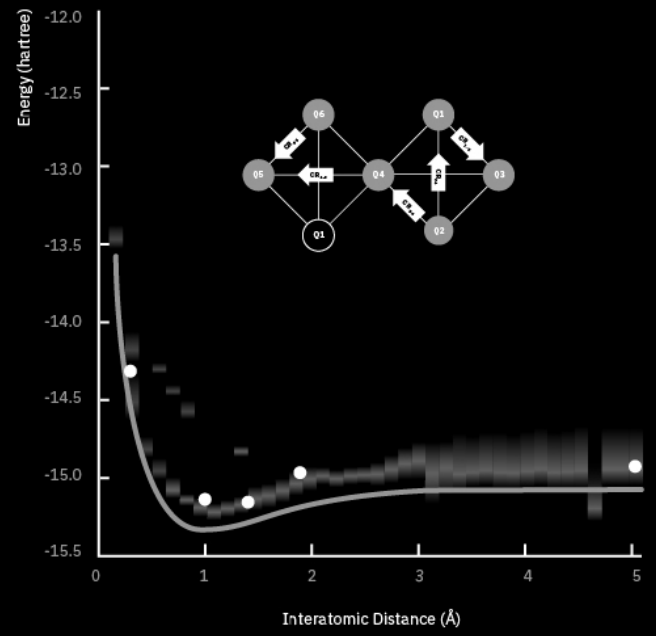




Lithium Hybride (LiH): 4 Qubits



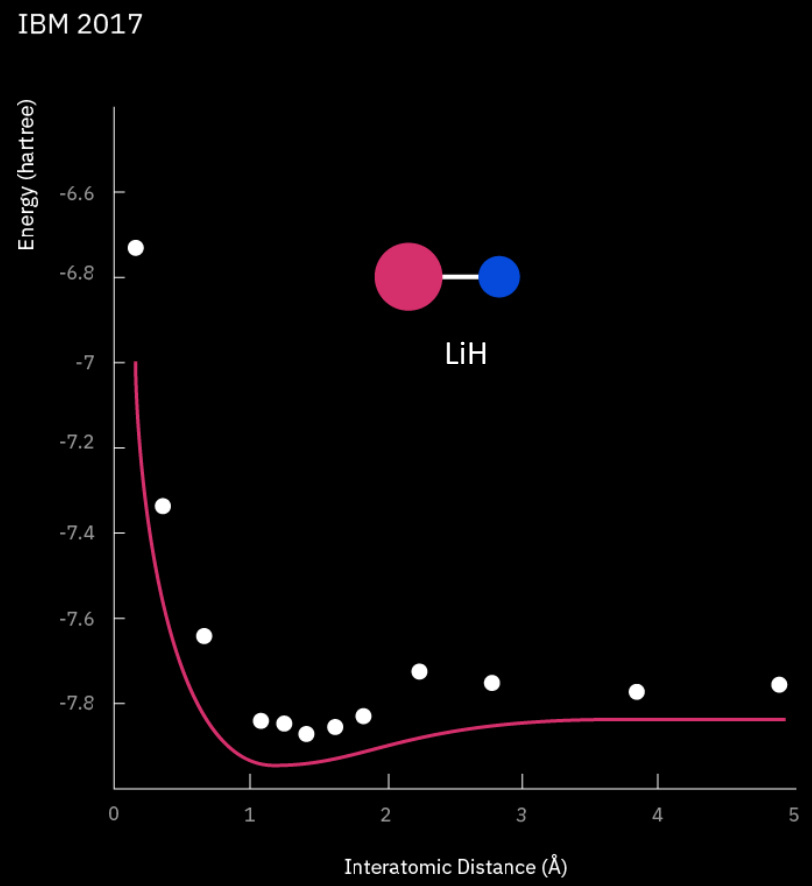
Beryllium hydride (BeH_2): 6 Qubits



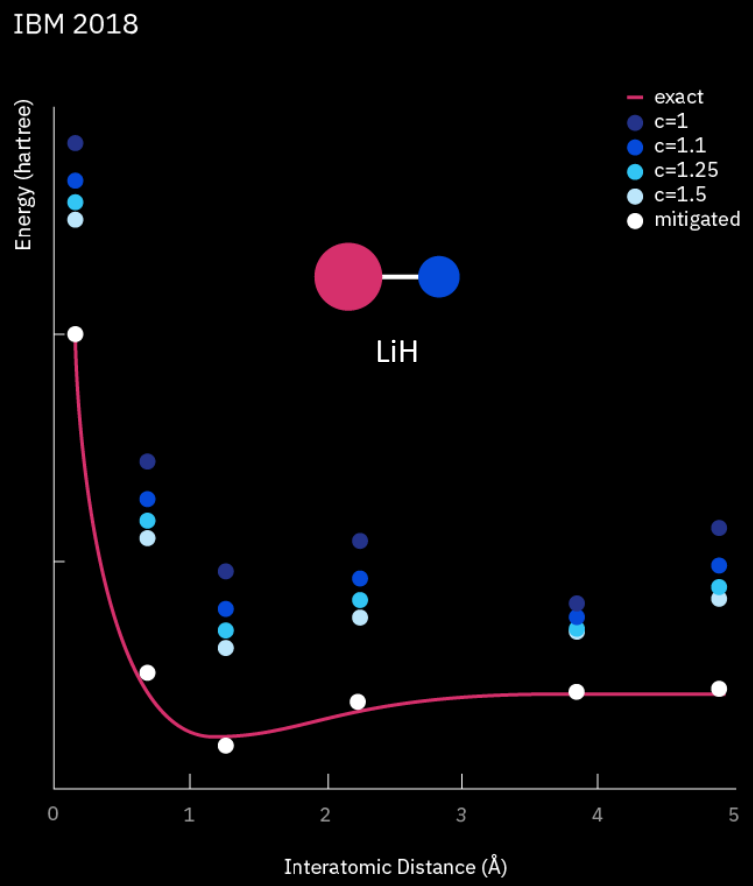
Error contributions from:

- Limited entangling steps
- Decoherence
- Sampling error
- Accuracy of classical optimizer

Accuracy improvement by integrating error mitigation



A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, J. M. Gambetta
Hardware-efficient variational quantum eigensolver for small molecules & quantum magnets
Nature **549**, 242, 2017



A. Kandala, K. Temme, A. D. Corcoles, A. Mezzacapo, J. M. Chow, J. M. Gambetta
Extending the computational reach of a noisy superconducting quantum processor
arXiv:1805.04492, 2018
Nature, **567**, 491, 2019

Quantum chemistry

Where we are Today

Studied **ground state** and **excited properties** of small molecules

New **hardware efficient ansatz**

Implemented **error mitigation** techniques

Impacts

Naturally suitable for quantum computers

Polynomial scaling in required resources vs exponential scaling for classical algorithms

Path to Quantum Advantage

More complex molecular systems with **50-100 qubits**

Implementation of **highly corelated electronic systems** which cannot be solved classical due to sign problem

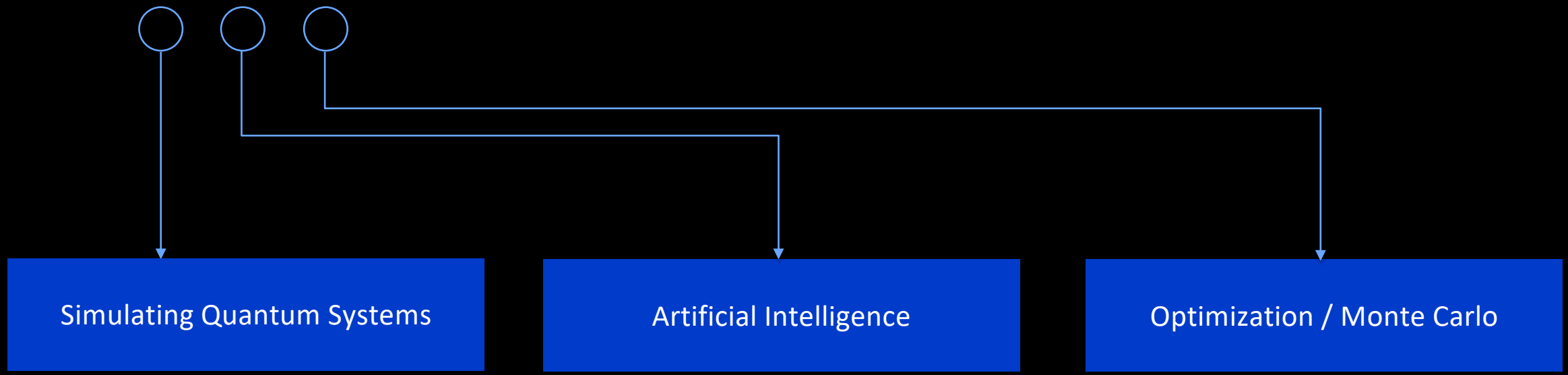
Challenges

Current **polynomial scaling** is still **very expensive**

Increase system sizes for simulations as well as hardware implementations

Reduce errors due to cross-talk and decoherence on the hardware

In collaboration with IBM Q Network partners, we are driving advancements in quantum software and algorithms.

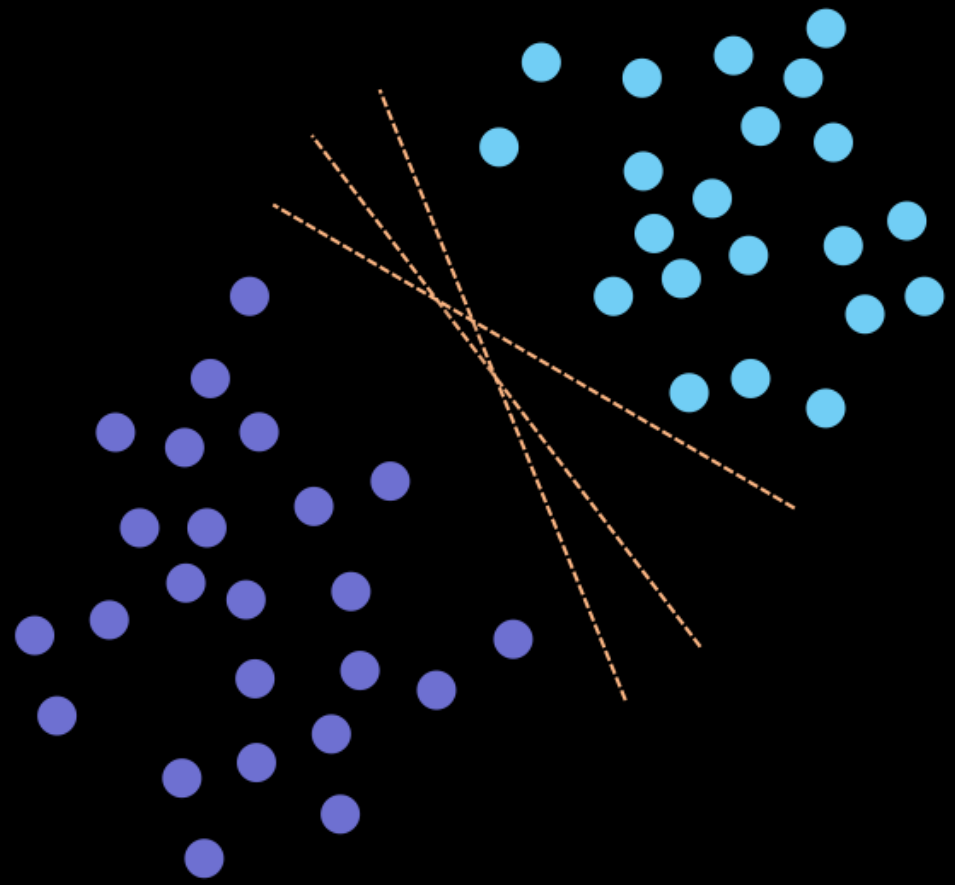


The problem we are considering is the classification of data points into two classes.

Suppose we want to assign labels using the following data.

By **drawing a line** separating the classes, we can assign labels to new data points based on which side of the line they are found.

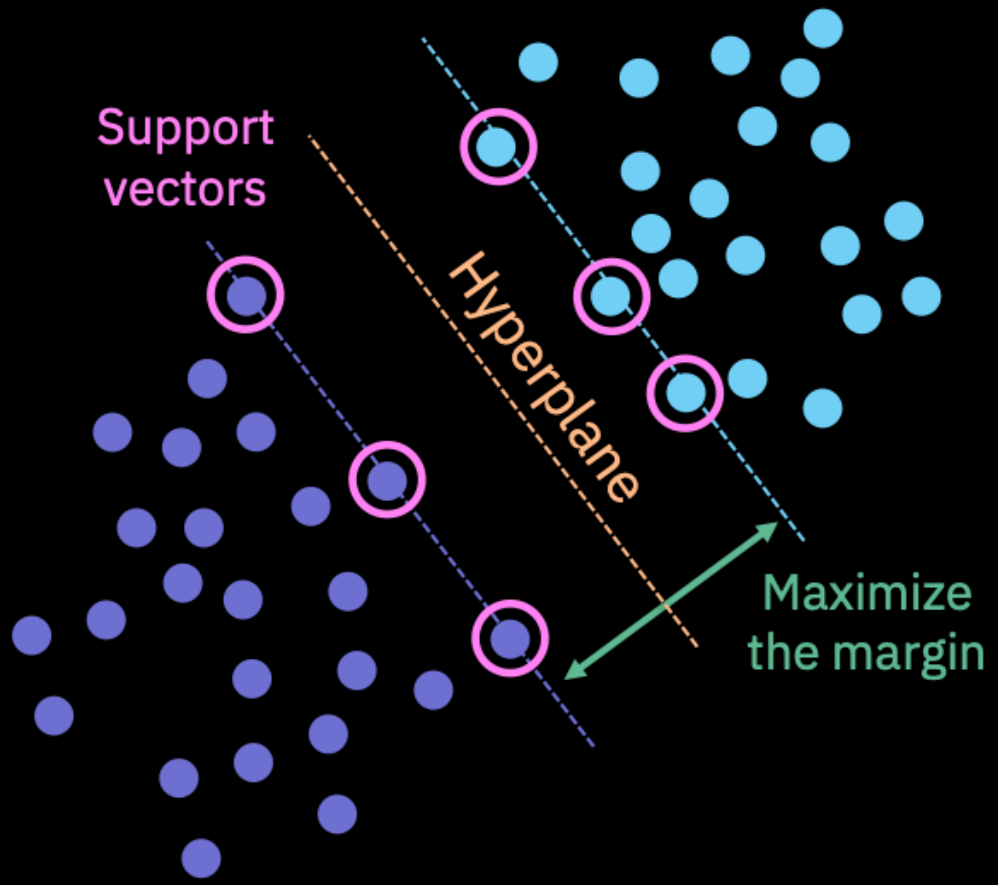
However, many different lines will cleanly separate the two classes.



We want the optimal line, or **hyperplane**, that corresponds to the **maximum margin**.

Then, we only need the **support vectors** from the training set to assign labels to new data points.

This is the task of a **support vector machine**.

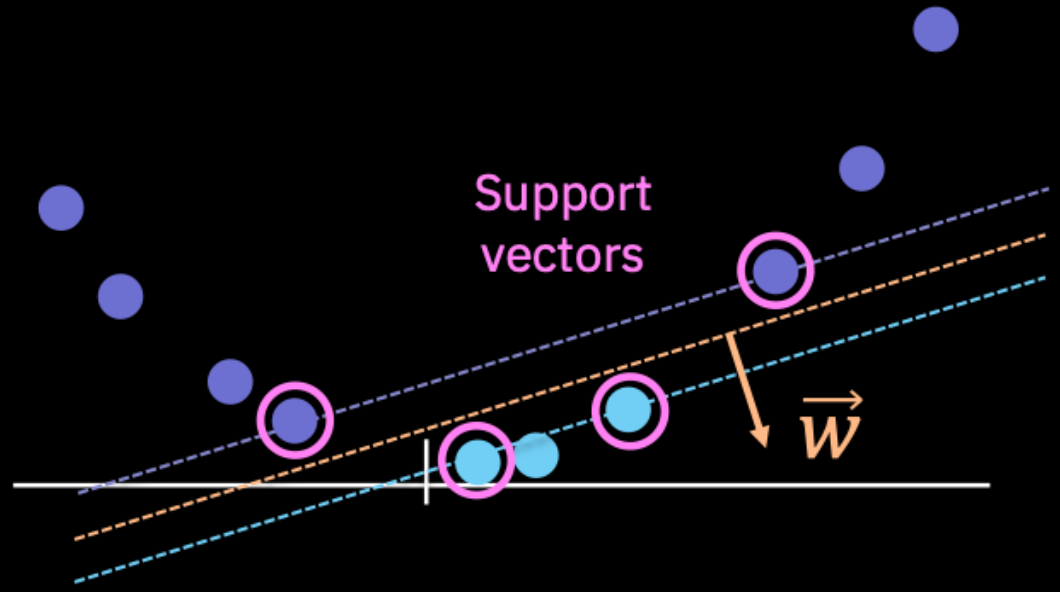


IBM

Often, the data are not linearly separable.

Non-linearly separable datasets may become linearly separable by **increasing dimensionality**.

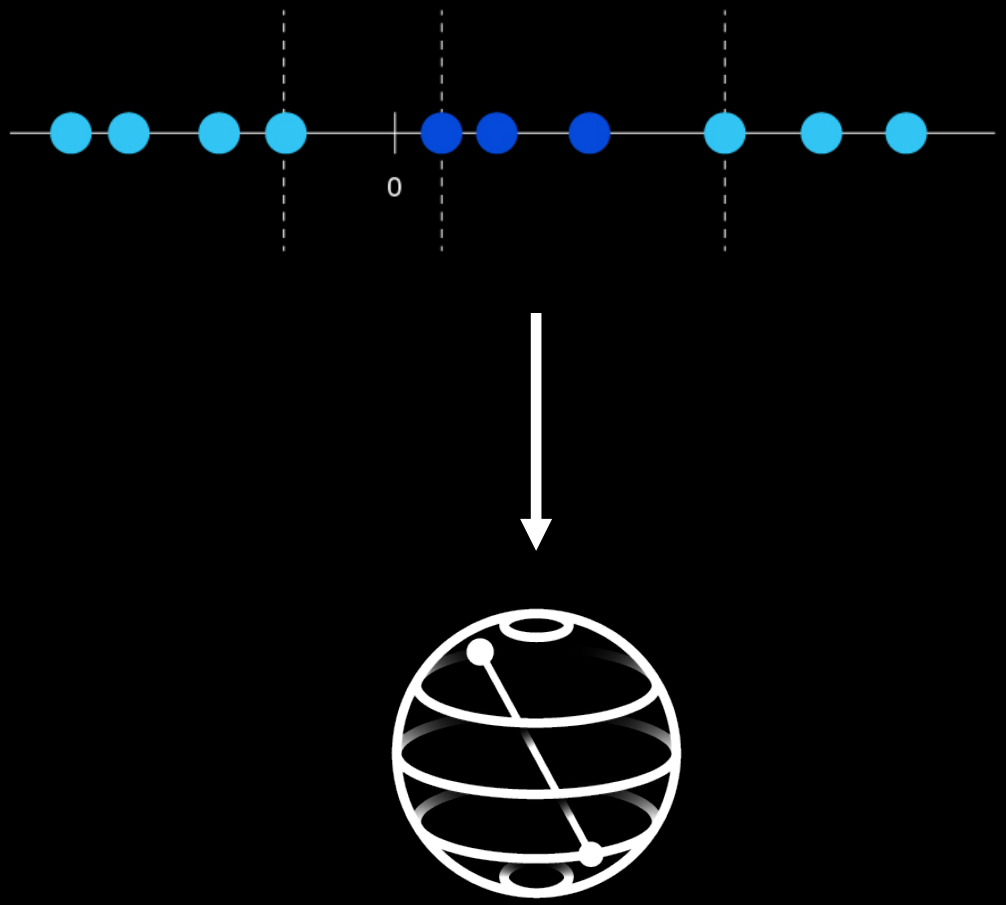
This transformation is called a **feature map** $\Phi(\vec{x})$.

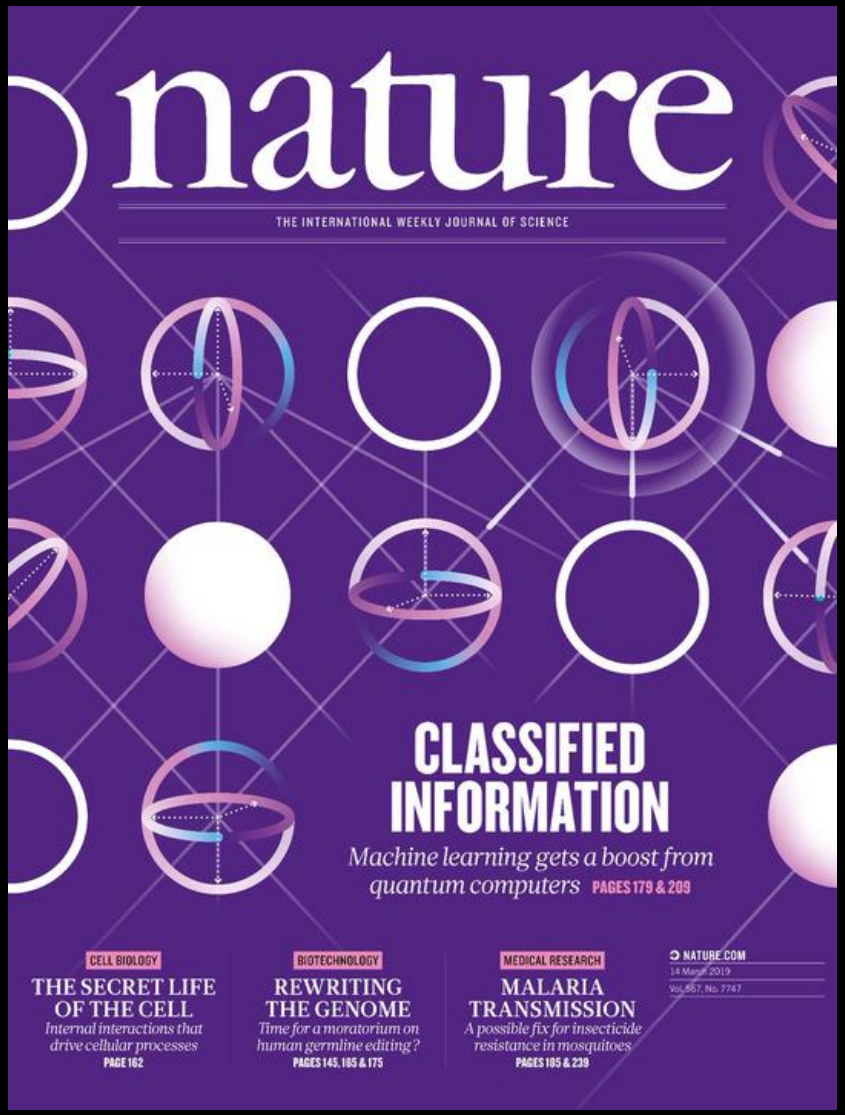


hyperplane $\vec{w} \cdot \Phi(\vec{x}) + b = 0$

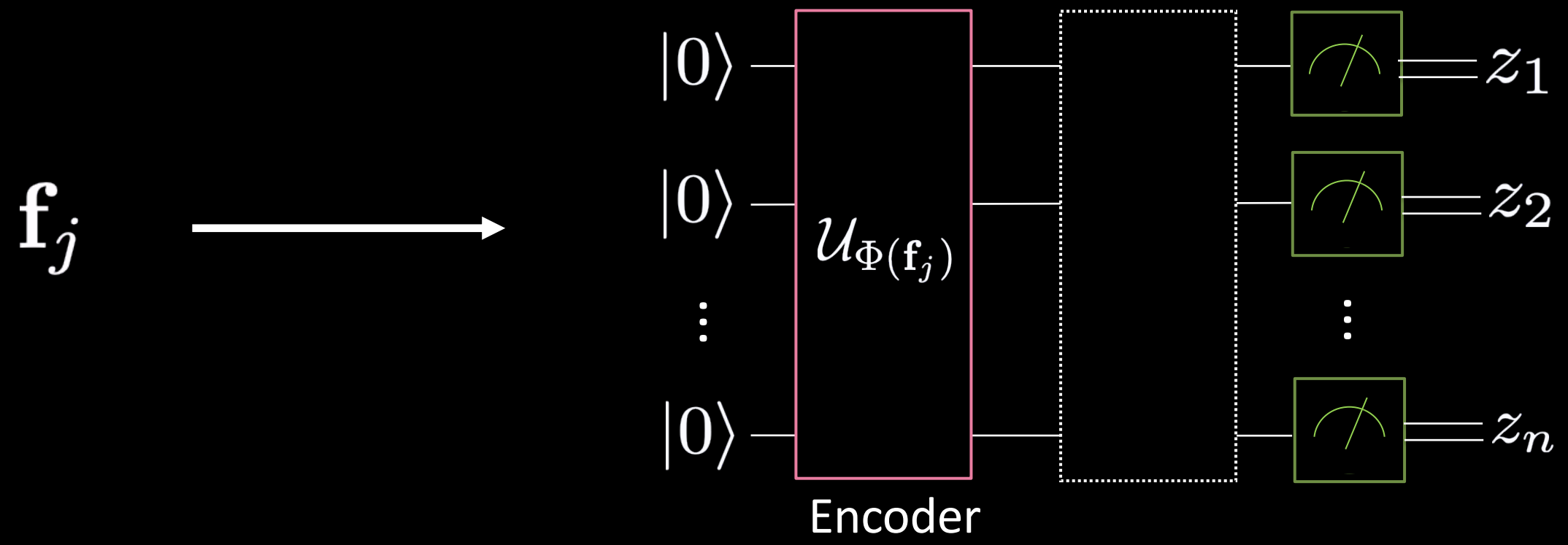
IBM

Improving classification with **quantum** feature engineering



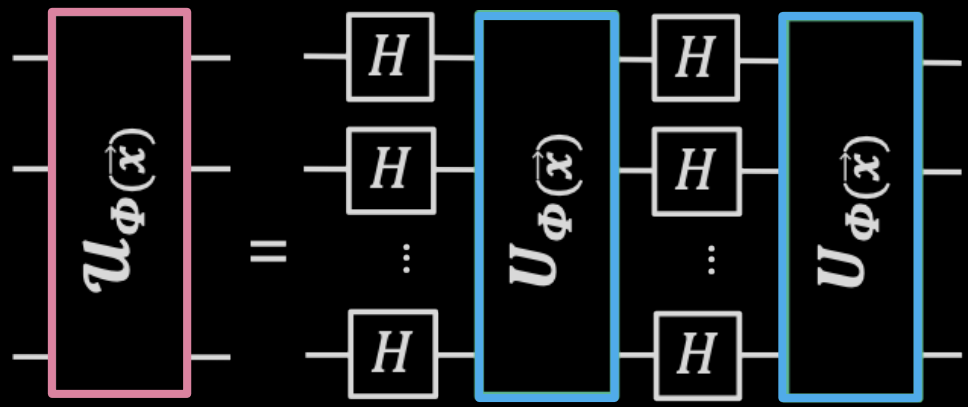


Encoding data in qubits



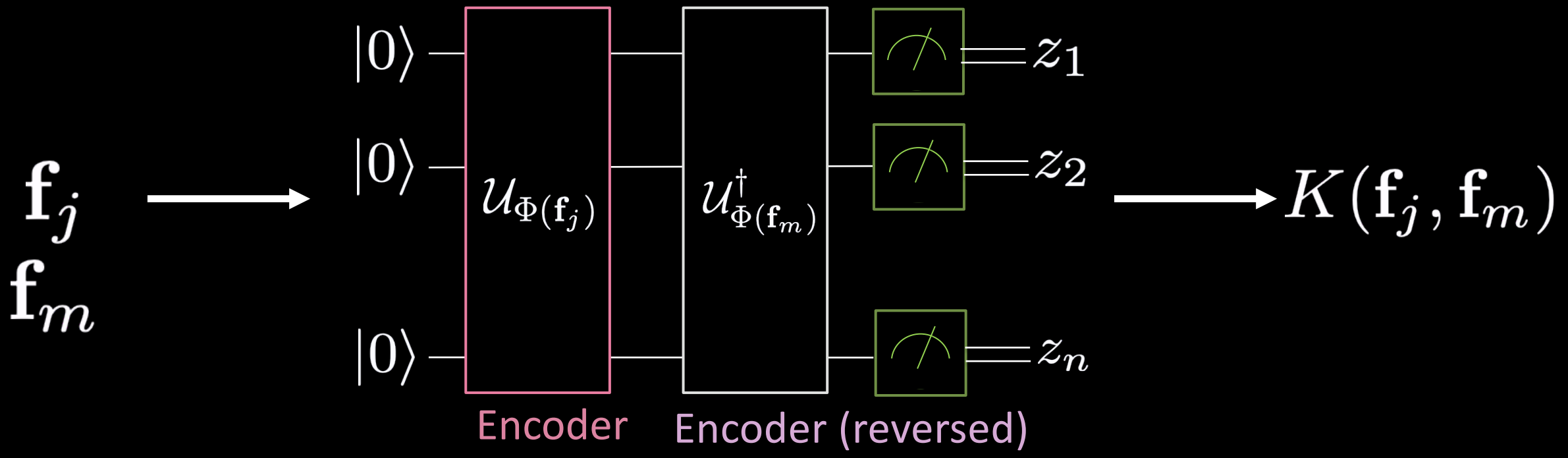
Entangling blocks encode data

$$U_{\Phi(\vec{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{j \in S} Z_j \right)$$



Using data encoded in qubits:
quantum kernel estimator

Compute quantum kernel elements, feed to classical ML



Quantum-enhanced feature spaces may offer advantages

over conventional support vector machines if the resulting kernel cannot be efficiently simulated classically.

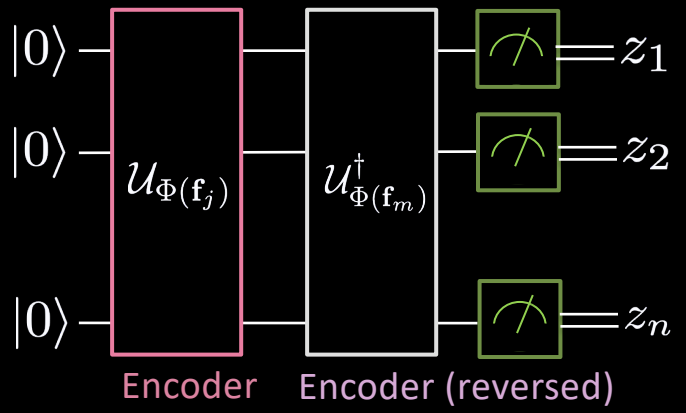
Example: short-depth quantum circuits as classifiers

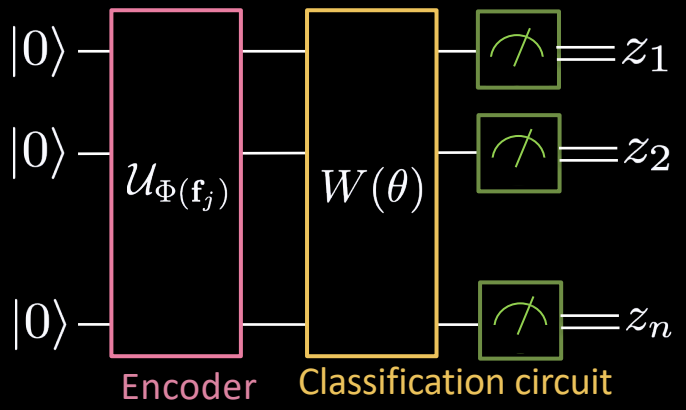
1

Function to learn

2

Quantum kernel or classifier





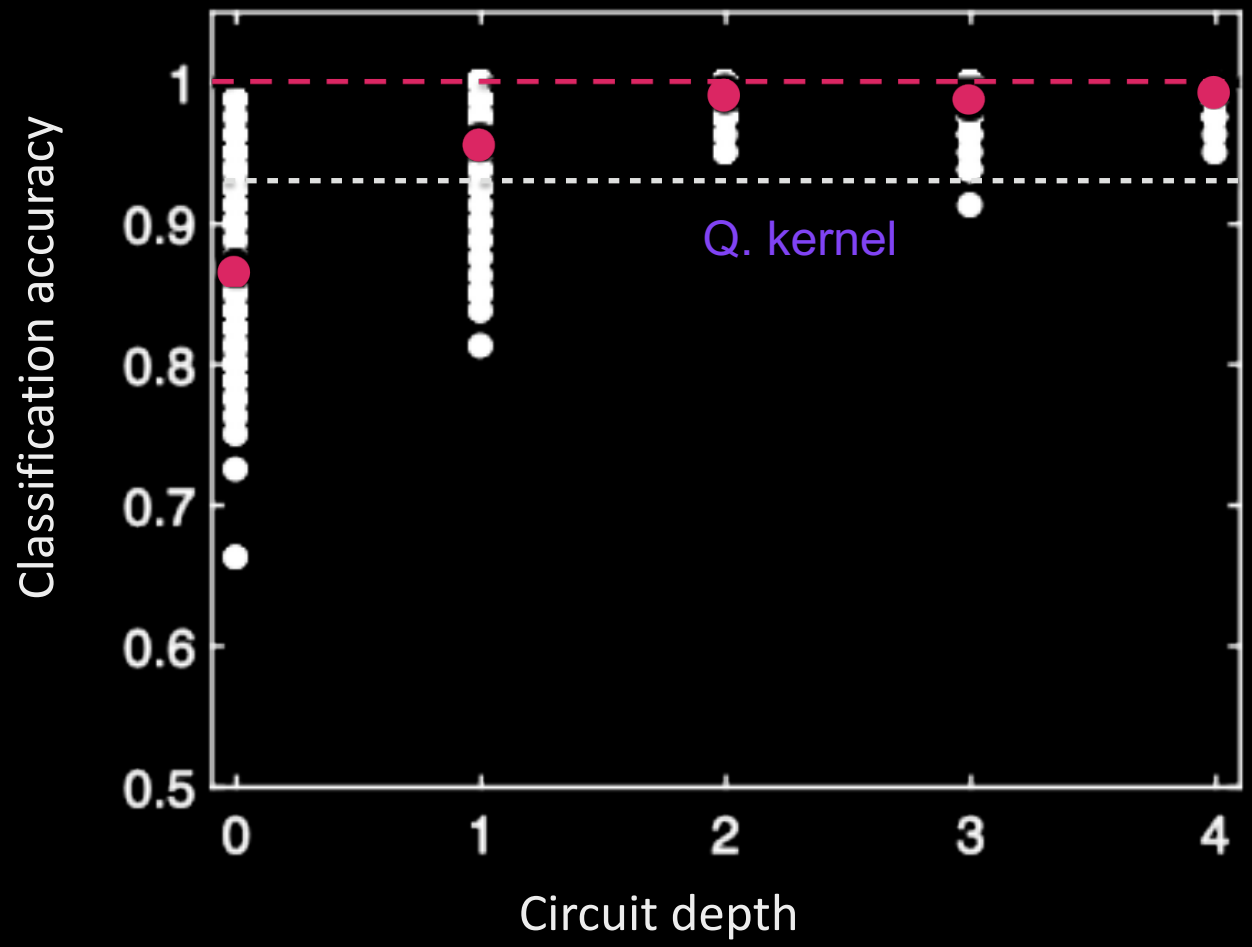
3

Assign label

black

gold

Classification accuracy increases with circuit depth.



Quantum machine learning

Impacts

New “**quantum kernels**”

New “**quantum classifiers**”

Status today

Implemented **binary classification** using a **quantum support vector machine**

Path ahead

New **feature maps**

Go to **higher-dimensional** data sets

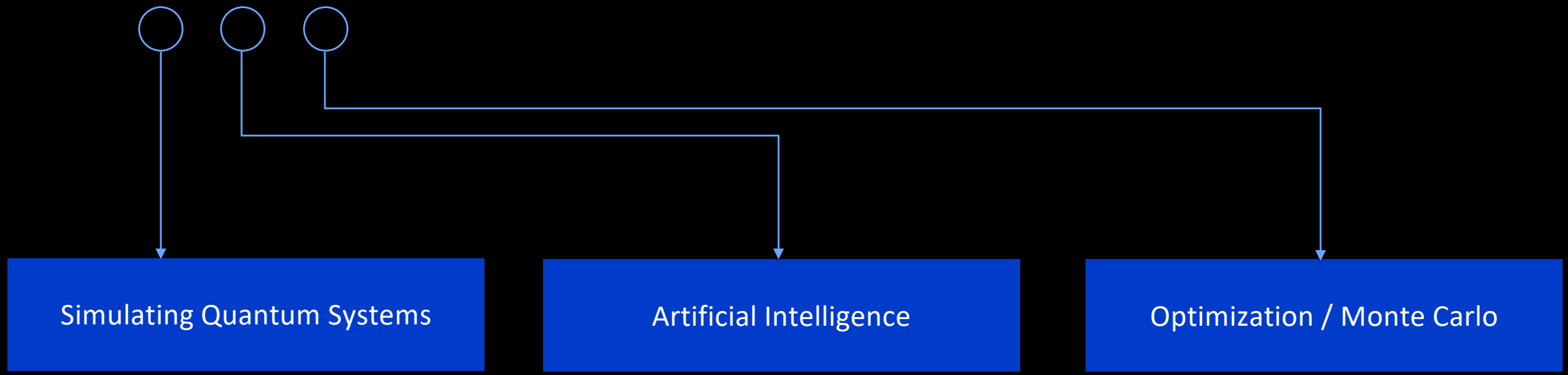
Extend to **more qubits**

Challenges

Improved **error mitigation** to scale up qubit count

Deploy on **practical data sets**

In collaboration with IBM Q Network partners, we are driving advancements in quantum software and algorithms.



Quantum Optimization problems

Cost function $C(\mathbf{z})$

Quantize: bits -> qubits $\{0,1\} \rightarrow |0\rangle, |1\rangle$

Cost -> *cost Hamiltonian* $C(\mathbf{z}) \rightarrow H_c$

Cost penalty -> energy of “cost Hamiltonian” $E(|\psi\rangle) = \langle\psi|H_c|\psi\rangle$

Solution = ground state of cost Hamiltonian

Near-term algorithms for combinatorial optimization

Variational eigensolver

$$E(|\psi(\theta)\rangle) = \langle\psi(\theta)|H_c|\psi(\theta)\rangle$$

Like chemistry! *No theoretical guarantees*

Quantum approximation optimization algorithm (QAOA)

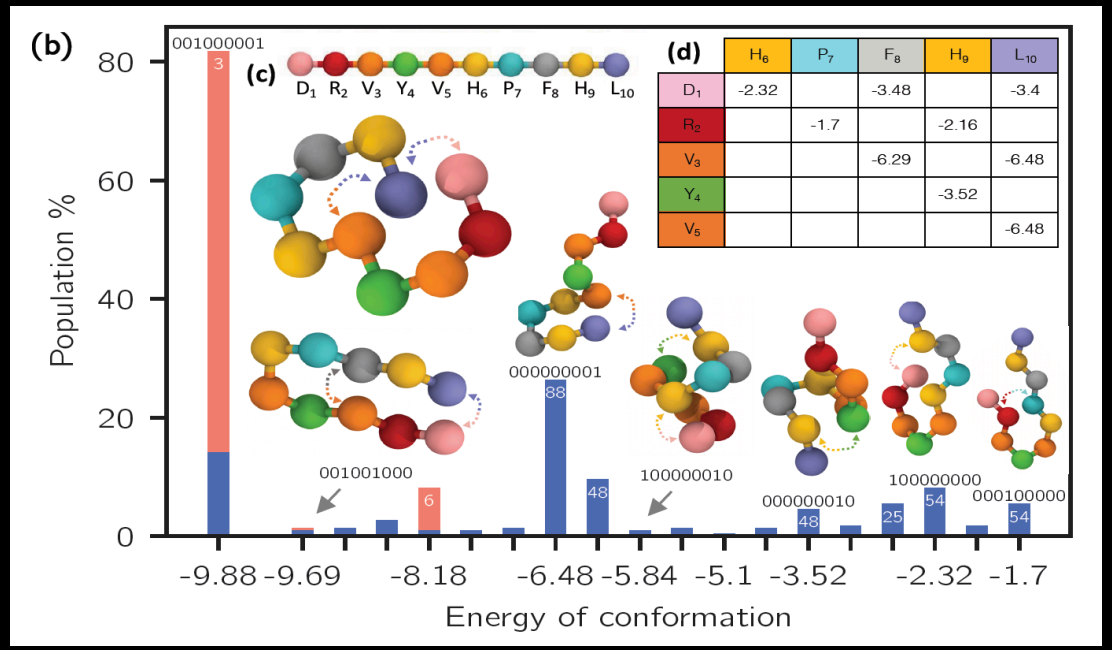
Make state $|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle = \prod_{j=1}^{s} e^{-i\beta_j H_a} e^{-i\gamma_j H_c} |+\rangle^{\otimes n}$

Add layers $s \to \infty \;:\; \langle\boldsymbol{\beta},\boldsymbol{\gamma}|H_c|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle \to \min_{\boldsymbol{z}} C(\mathbf{z})$

Theoretical guarantees! *Identify good value(s) for angles*

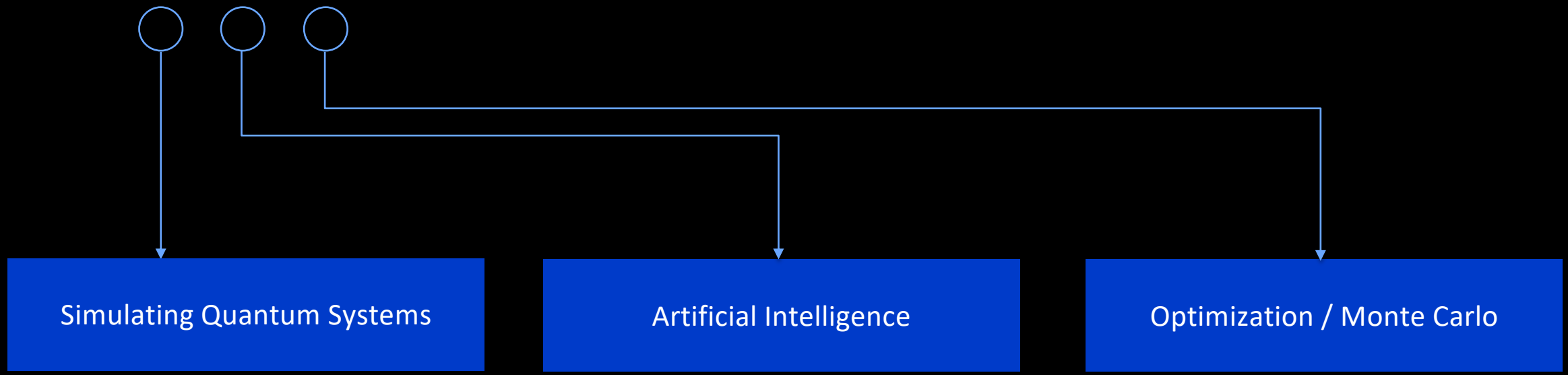
Protein folding as an optimization problem

- Recent publication of protein folding example on a quantum computer from IBM Zurich
- Protein monomers placed on tetrahedral grid
- Problem is defined in terms of configurational qubits and interaction qubits
- Hamiltonian encodes penalty terms based on configurational arrangements and interaction energy terms
- VQE based approach at finding low energy configurations



arXiv: 1908.02163v1

In collaboration with IBM Q Network partners, we are driving advancements in quantum software and algorithms.

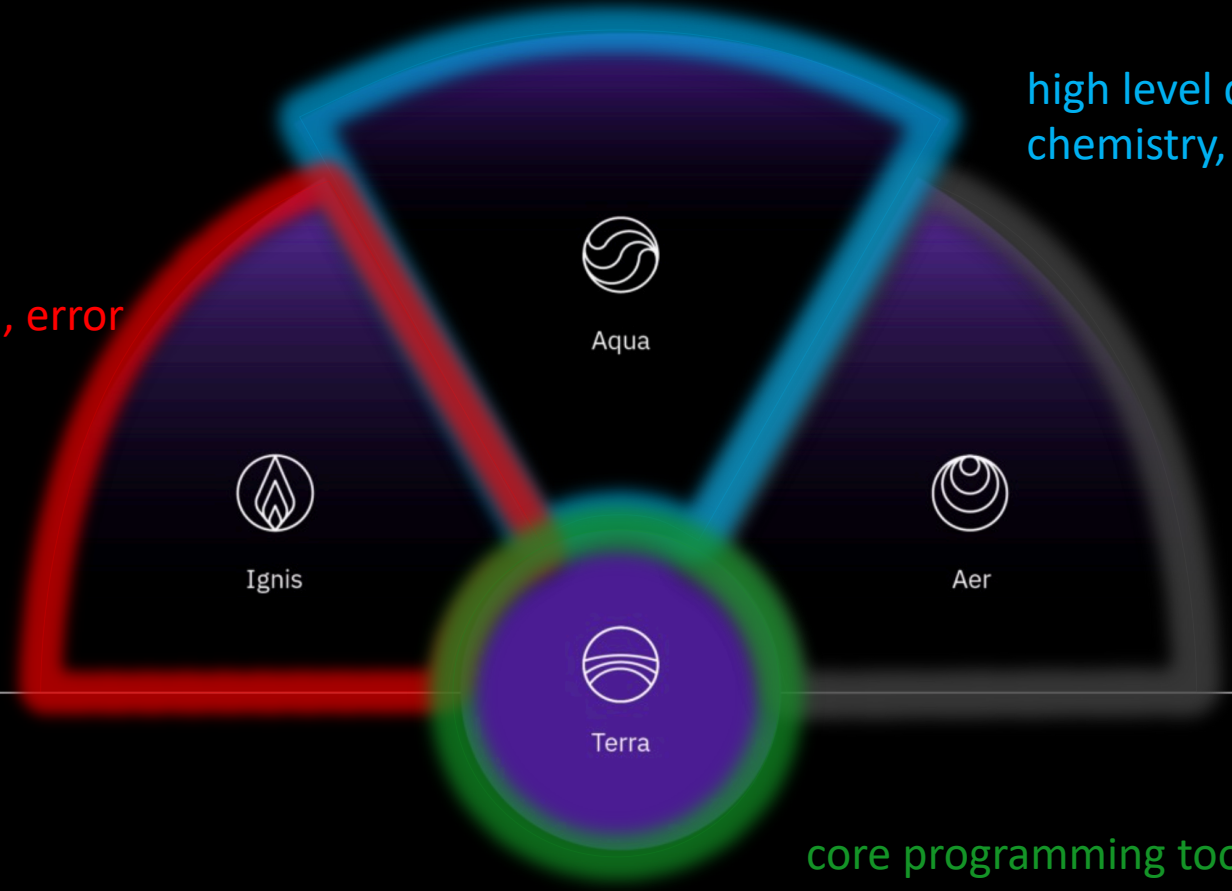


Qiskit

high level quantum applications:
chemistry, optimization, AI, finance

state characterization, error mitigation,
optimal control

classical simulation of quantum circuits



Qiskit

Hardware

core programming tools and API to hardware

Terra

Terra is a collection of core, foundational tools for communicating with quantum devices and simulators. Users can write quantum circuits, and address real hardware constraints with Terra. Its modular design simplifies adding extensions for quantum circuit optimizations and backends.

Ignis

Controlling fire was a turning point in human evolution. Learning how to fix or control quantum errors will be a turning point in the evolution of quantum computing. Users can access better characterization of errors, improve gates, and compute in the presence of noise with Ignis. It is designed for researching and improving errors or noise in near-term quantum systems.

Aqua

Aqua is a modular and extensible library for experimenting with quantum algorithms on near-term devices. Users can build domain-specific applications, such as chemistry, AI and optimization with Aqua. It bridges quantum and classical computers by enabling classical programming to run on quantum devices.

Aer

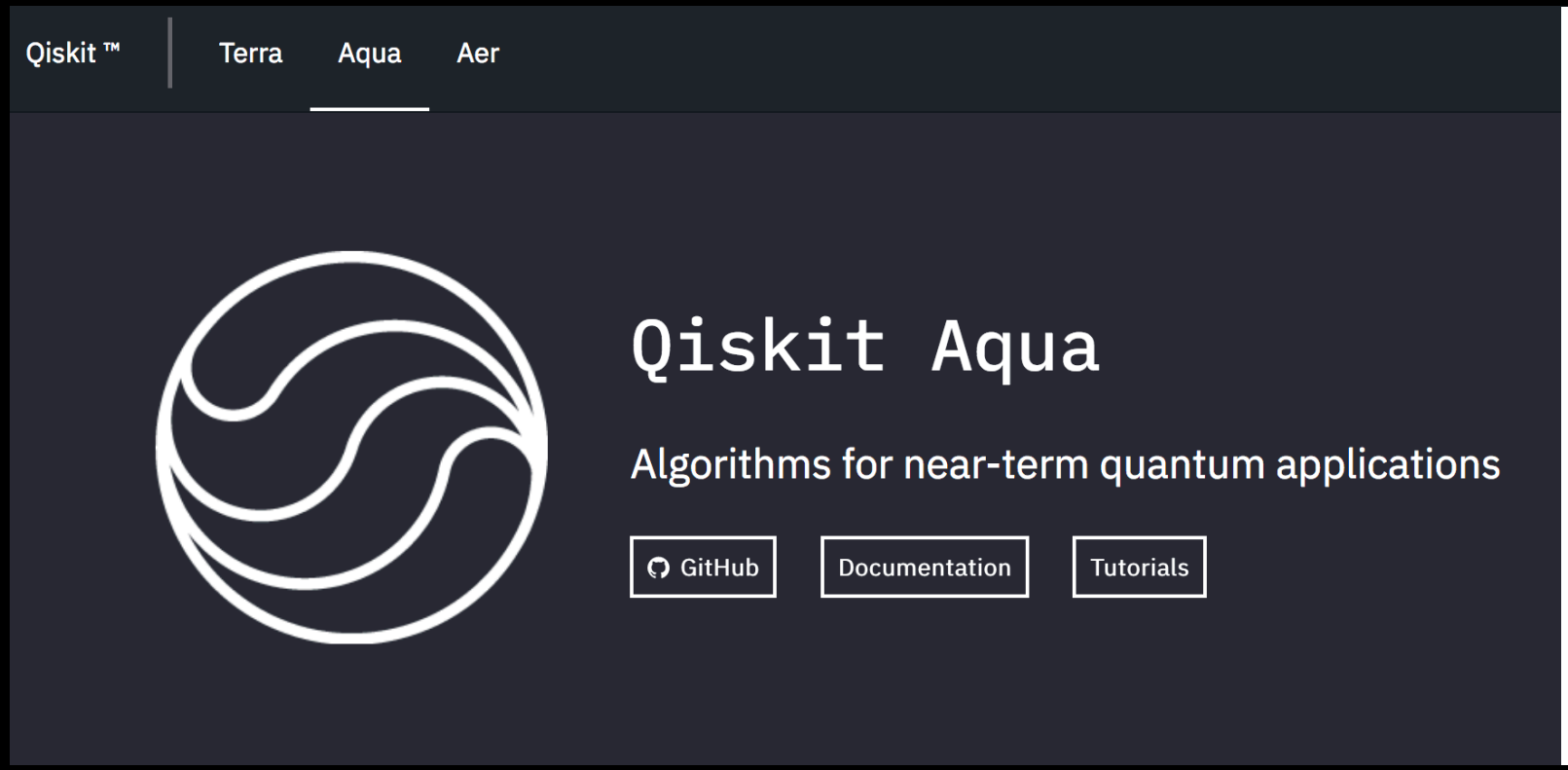
Aer permeates all other Qiskit elements. Users can accelerate their quantum simulator and emulator research with Aer, which helps to better understand the limits of classical processors by demonstrating their ability to mimic quantum computation. Users can also verify current and near-term quantum computer functionality with Aer.

Panagiotis Barkoutsos - bpa@zurich.ibm.com

Quantum chemistry with Qiskit Chemistry

https://qiskit.org/aqua

pip install qiskit-chemistry



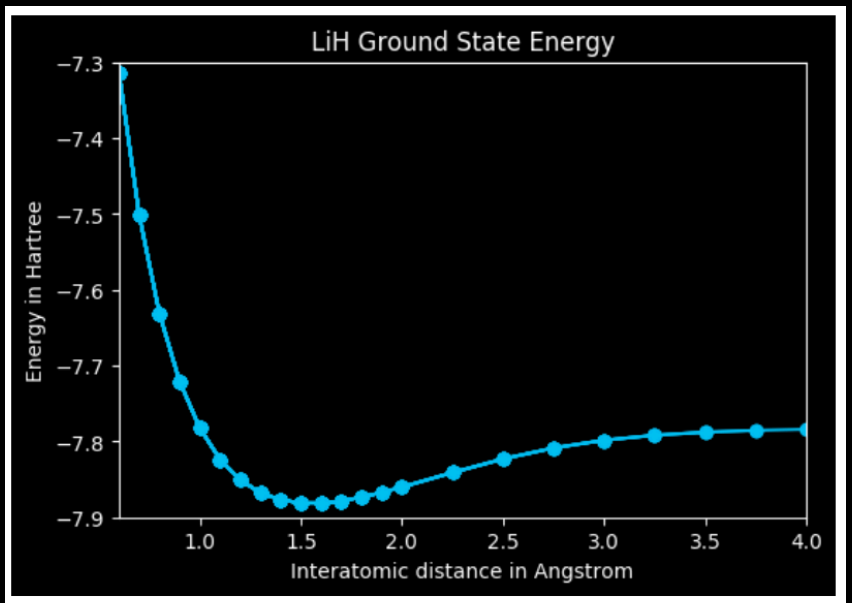
Qiskit Chemistry

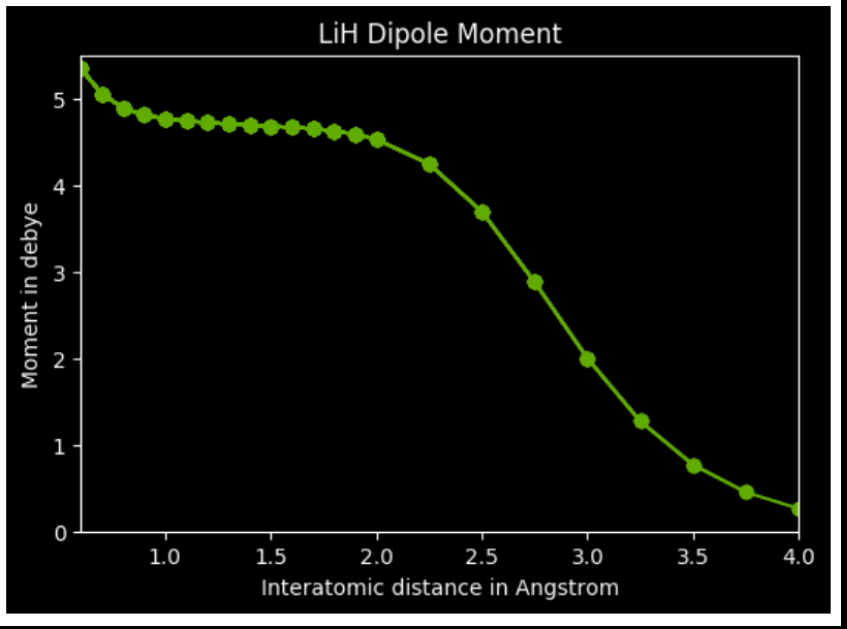
GitHub | Documentation | Tutorials

Qiskit Chemistry is the first end-to-end stack that enables experimenting with chemistry problems on near-term quantum computers. It translates chemistry-specific problems defined via classical drivers into inputs for Aqua algorithms. It is modular and extensible, and allows users with different levels of experience to execute chemistry experiments and contribute to the quantum computing chemistry software stack.

Calculating ground state molecular properties with Qiskit Chemistry

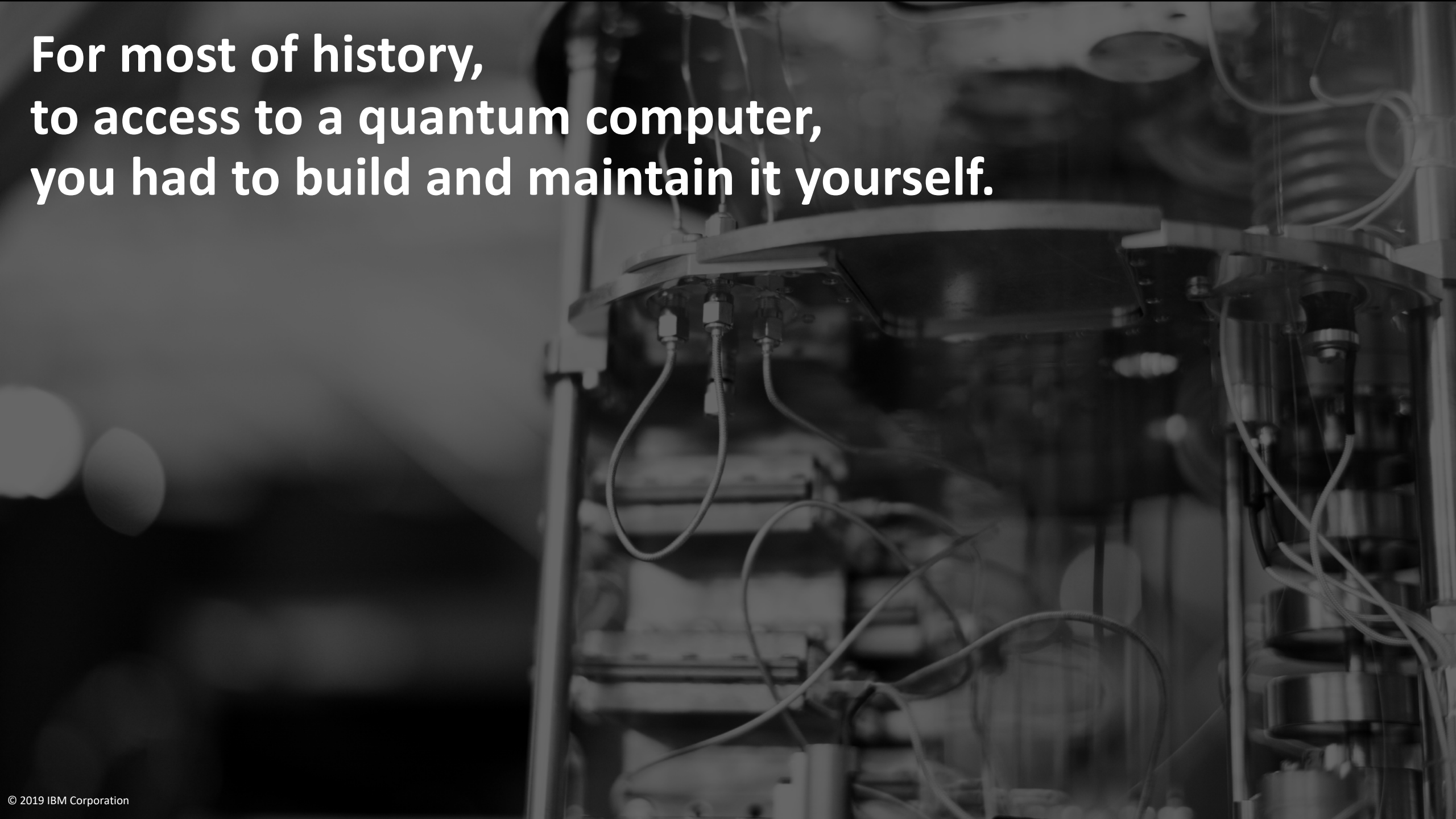
```python
import numpy as np
from qiskit_chemistry import QiskitChemistry

qiskit_chemistry_dict = {
  "driver": { "name": "PYSCF" },
  "PYSCF": { "atom": "", "basis": "sto3g" },
  "operator": {
    "name": "hamiltonian",
    "qubit_mapping": "parity",
    "two_qubit_reduction": True,
    "freeze_core": True,
    "orbital_reduction": [-3, -2]
  },
  "algorithm": { "name": "VQE" },
  "optimizer": { "name": "COBYLA", "maxiter": 10000 },
  "variational_form": { "name": "UCCSD" },
  "initial_state": { "name": "HartreeFock" }
}
molecule = "H .0 .0 -{0}; Li .0 .0 {0}"

pts  = [x * 0.1  for x in range(6, 20)]
pts += [x * 0.25 for x in range(8, 16)]
pts += [4.0]
energies = np.empty(len(pts))
distances = np.empty(len(pts))
dipoles = np.empty(len(pts))

for i, d in enumerate(pts):
  qiskit_chemistry_dict["PYSCF"]["atom"] = molecule.format(d/2)
  solver = QiskitChemistry()
  result = solver.run(qiskit_chemistry_dict)
  energies[i] = result["energy"]
  dipoles[i] = result["total_dipole_moment"] / 0.393430307
  distances[i] = d

for j in range(len(distances)):
  print("{:0.2f}: Energy={:0.8f}, Dipole={:0.5f}".format(distances[j]
```





IBM Q System One

Naked Quantum Computer



Three years ago, that changed.

IBM

May 3, 2016
Posted in: Quantum Computing

Quantum computing: It's time to build a quantum community

WIRED

CADE METZ BUSINESS 05.04.16 12:01 AM

IBM IS NOW LETTING ANYONE PLAY WITH ITS QUANTUM COMPUTER

The New York Times

IBM Wants Everyone to Try a Quantum Computer

By John Markoff

May 4, 2016

TechCrunch

IBM launches quantum computing as a cloud service

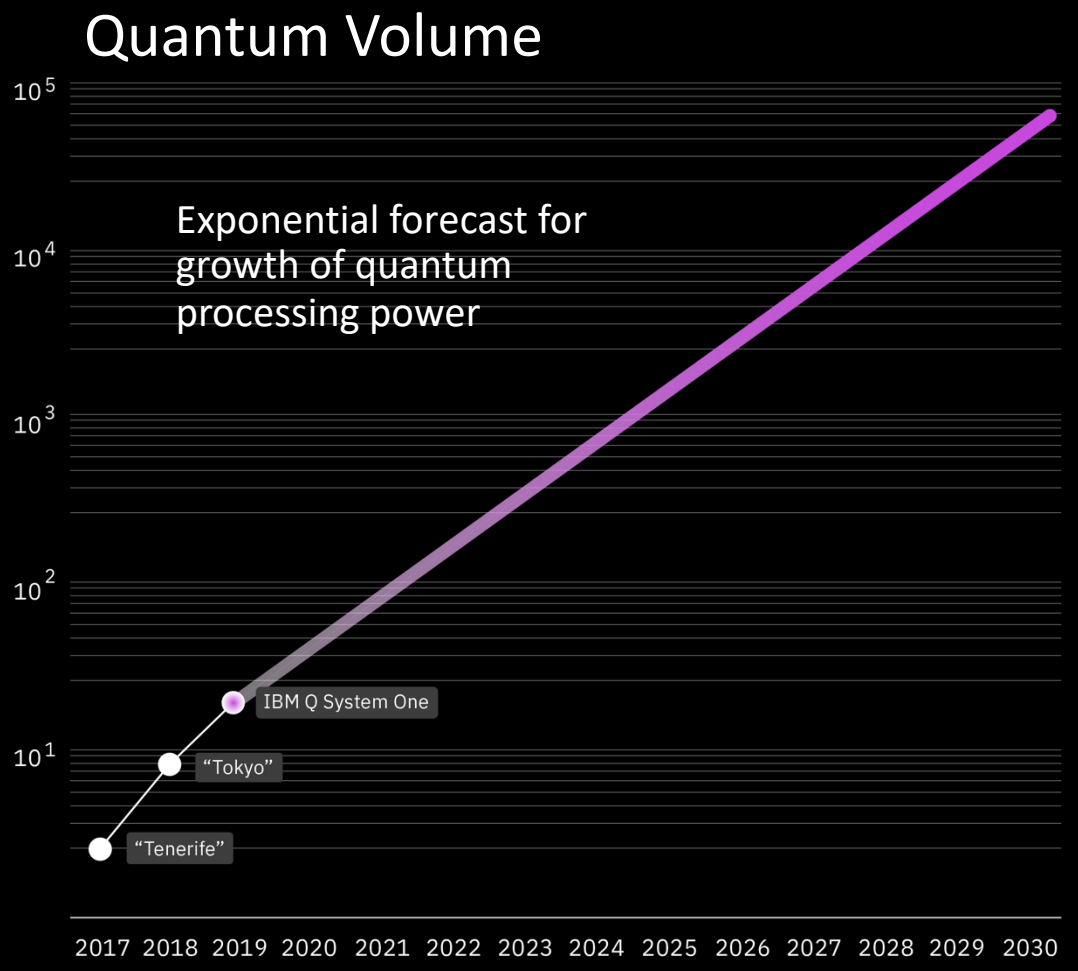
Ron Miller @ron_miller / May 4, 2016

Our quantum systems roadmap is based on three metrics.

Functionality:
Double quantum volume every year

Performance:
Continue to reduce average runtime per job; enable increased throughput

Availability:
Continued focus on uptime via improved system management features and continued system additions



"The quantum volume is a pragmatic way to measure and compare progress toward improved system-wide gate error rates for near-term quantum computation and error-correction experiments."

Cross, Bishop, Sheldon, Nation, and Gambetta, arXiv:1811.12926

IBM is committed to building a thriving open-source community.

Qiskit

Collaboratively advances quantum science with open-source tools and capabilities in a modular framework.

IBM Quantum Experience

Provides quantum cloud services, tools, and capabilities optimized for IBM Q systems.

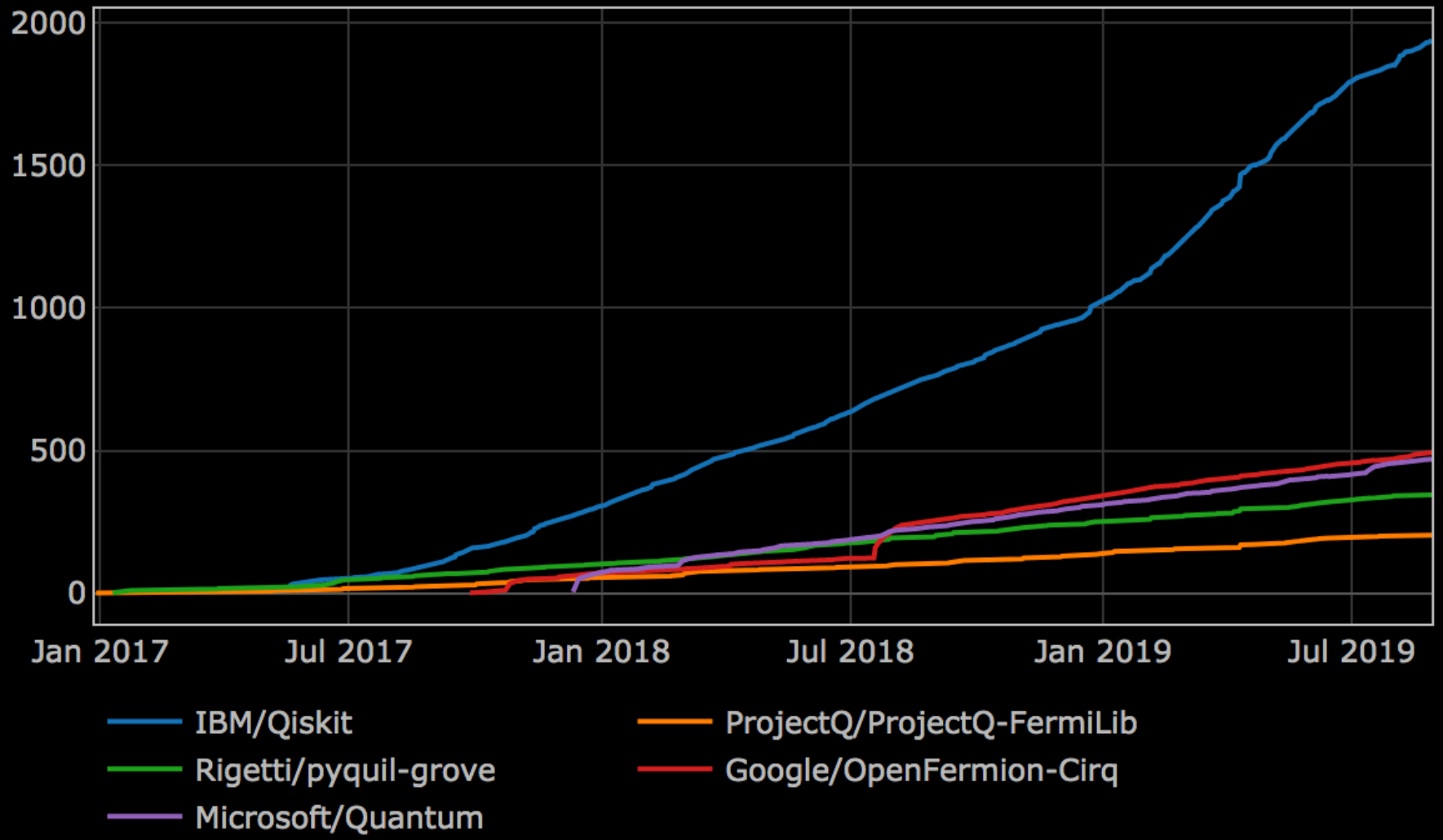
Qiskit community

Qiskit is driven by our avid community of Qiskitters! We are committed to our goal of bringing quantum computing to people of all backgrounds, and are always excited to hear your feedback directly from you. There are many ways to stay informed, contribute to, and collaborate on Qiskit.

- Slack
- GitHub
- Stack Exchange
- Twitter
- Medium
- YouTube
- Facebook

Qiskit is the most feature-rich, and popular, software framework.

GitHub Forks (number of developers using/contributing to Open Source)



IBM Q Network

102 members

- 10 industry partners
- 13 hubs
- 20 industry and research members
- 28 startups
- 31 academic partners

Industry Partners

JP Morgan Chase & Co.

ExxonMobil

Samsung

Daimler

JSR Corp

Accenture

Goldman Sachs

Woodside Energy

BP

Amgen

Hubs

Japan-IBM Quantum Partnership

German Hub at Fraunhofer

US Air Force Research Lab

Oak Ridge National Lab

Keio University

NC State

Munich Hub at U. Bundeswehr

National Taiwan University

Iberian Nanotech Lab

CSIC Spain

Los Alamos National Laboratory

U. Melbourne

U. Oxford

Members

Delta

Anthem

Wells Fargo

Barclays

Mizuho

MUFG

Mitsubishi Chem

Argonne Lab

Fermilab

Berkeley Lab

Brookhaven Lab

Naval Res Lab

ITRI

III Taiwan

CERN

Quemix

U. Minho

Honda

Hitachi Metals

Nagase

Startups

CQC

QC Ware

Grid

1QBit

Zapata

Strange Works

Q-CTRL

Quantum Benchmark

MDR

Qu&Co

JoS Quantum

SolidStateAI

ProteinQure

Labber Quantum

MaxKelsen

Netramark

Entropica

Boxcat

Rahko

Qunasys

QuantFi

Agnostiq

Tradeteq

AIQTech

Zurich Instruments

BEIT

Quantum Machines

A*Quantum

Academic

MIT

EDX.org

Virginia Tech

U. Montpellier

Notre Dame

Harvard

Princeton

Florida State

U. Stony Brook

U. Chicago

Duke

CU Boulder

U. Waterloo

U. Illinois

Northwestern

NYU

Wits

Aalto University

U. of Turku

U. Basque Country

U. of Innsbruck

EPFL

Chalmers University

ETH Zurich

Saarland University

Johns Hopkins

Boston University

U. Automata Madrid

Purdue

Stanford

Georgia Tech

IBM Q Network

- Industry Partners
- Hubs
- Members
- Startups
- Academic Partners