Quantum Computing: Quantum Gates

Rotation Gates:

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

$R_X(\pi) = X$, $R_Y(\pi) = Y$, $R_Z(\pi) = Z$, $R_Z(\pi/2) = S$, $R_Z(\pi/4) = T$

Quantum Computing: Quantum Gates

Tensor Product of One-qubit Gate:

The simplest way of obtaining a two-qubit gate is by having a pair of one-qubit gates A and B acting on each of the qubits

$$(A \otimes B)(|\psi_1\rangle \otimes |\psi_2\rangle) = (A|\psi_1\rangle) \otimes (B|\psi_2\rangle)$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \otimes \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{1,2}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \\ a_{2,1}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{2,2}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{bmatrix}$$

Quantum Computing: Quantum Gates

Controlled-NOT (CNOT) gate:

The CNOT (or controlled-NOT or C_X) gate is given by the (unitary) matrix

$$C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

It uses convention qubits labelling: $|q_1 q_0\rangle$, i.e. |second qubit, first qubit>

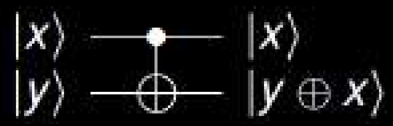
If the first qubit is 0, then nothing changes, otherwise flip the second qubit:

$$|00\rangle \rightarrow |00\rangle, \quad |01\rangle \rightarrow |11\rangle, \quad |10\rangle \rightarrow |10\rangle, \quad |11\rangle \rightarrow |01\rangle$$

For convention $|q_0 q_1\rangle$, i.e. |first qubit, second qubit>, then

$$C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Its action on $x, y \in \{0, 1\}$ is, then:



Quantum Computing: Entanglement

A state $|\psi\rangle$ is a product state if it can be written in the form

$$|\psi\rangle = |\psi_1\rangle|\psi_2\rangle$$

where $|\psi_1\rangle$ and $|\psi_2\rangle$ are two states (of at least one qubit).

An **entangled** state is a state that is **not** a product state (cannot be factored).

Example of entangled states are **Bell states**:

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$\frac{|01\rangle + |10\rangle}{\sqrt{2}} \qquad \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Quantum Computing: Decoherence

Decoherence is the interactions of a qubit with its environment which causes disturbances and collapse superposition.

The decoherence of a quantum system[1] is caused by thermodynamically irreversible interactions with the environment; it represents the principal mechanism for the transition from quantum to classical behavior. The evolution of a quantum system in contact with its environment is characterized by various decoherence times; each decoherence time is related to a different degree of freedom of the system.

The decoherence times relevant for a quantum computer are associated with the degrees of freedom that characterize the physical qubits; they also depend on the specifics of the qubits' couplings to these degrees of freedom[1].

Some scientists achieved coherence time exceeds one hour[2].

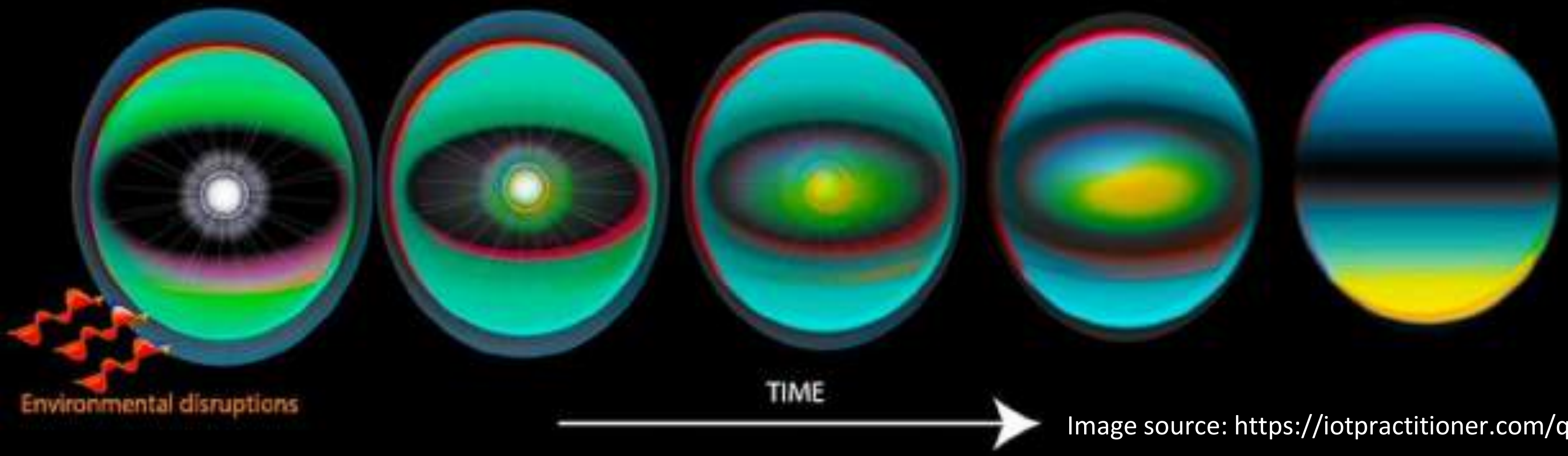


Image source: https://iotpractitioner.com/quantum-computing-series-part-8-decoherence

1. Dan C. Marinescu, Gabriela M. Marinescu, *Classical and Quantum Information*, (2012)

2. Wang, P., Luan, CY., Qiao, M. et al. *Single ion qubit with estimated coherence time exceeding one hour*. Nat Commun 12, 233 (2021)

Quantum Computing: Decoherence

Quantum Error Correction[1] can be used to prolong coherence length by correcting errors caused by decoherence.

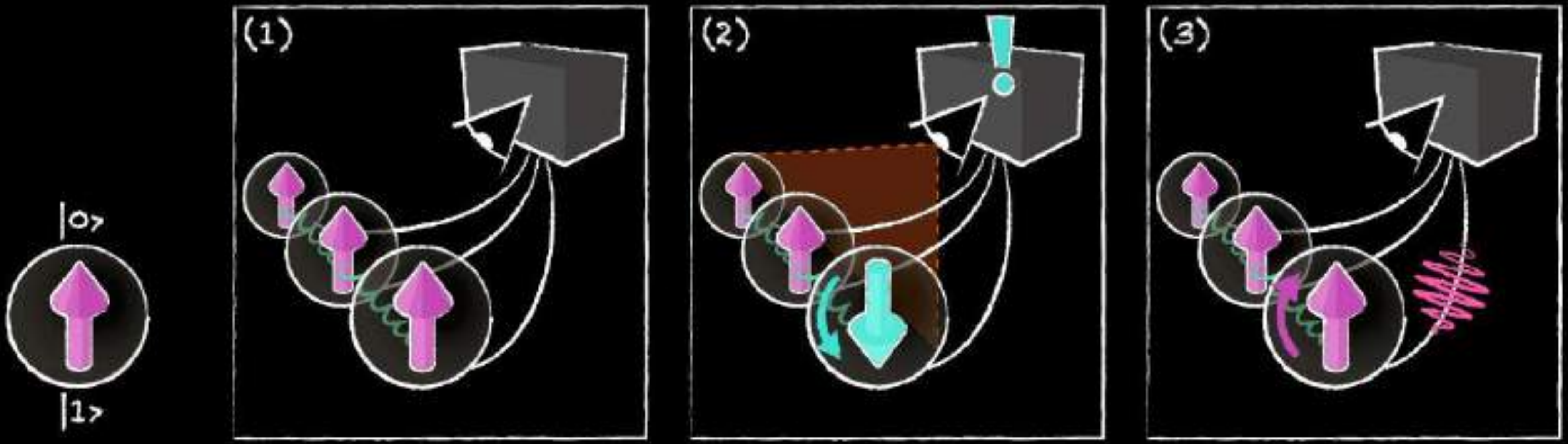
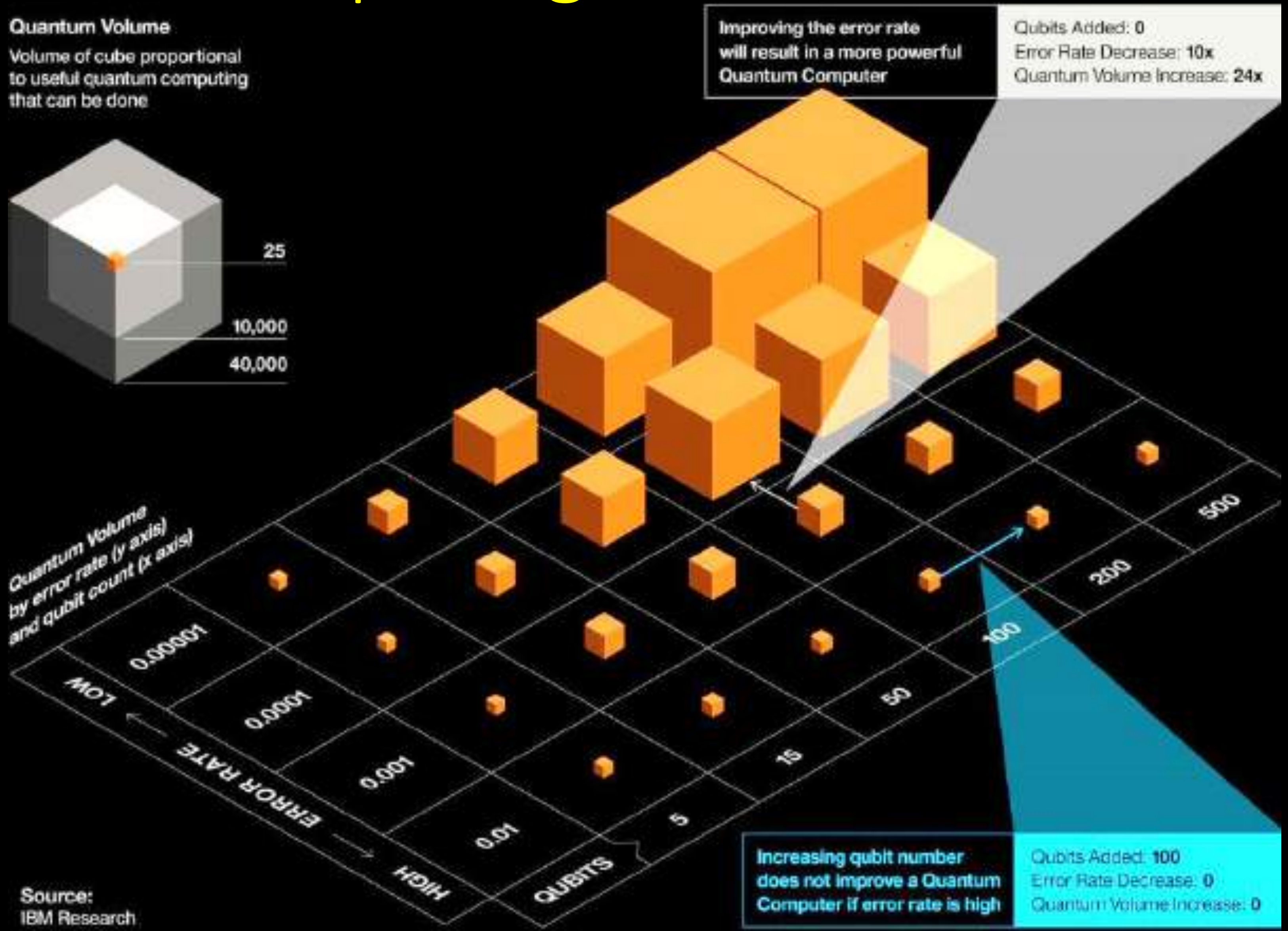


Image sourcehttps://medium.com/hackernoon/decoherence-quantum-computers-greatest-obstacle-67c74ae962b6

1. Simon J Devitt et al., *Quantum error correction for beginners*. Rep. Prog. Phys. 76 076001 (2013)

Quantum Computing: Quantum Volume



Quantum Computing

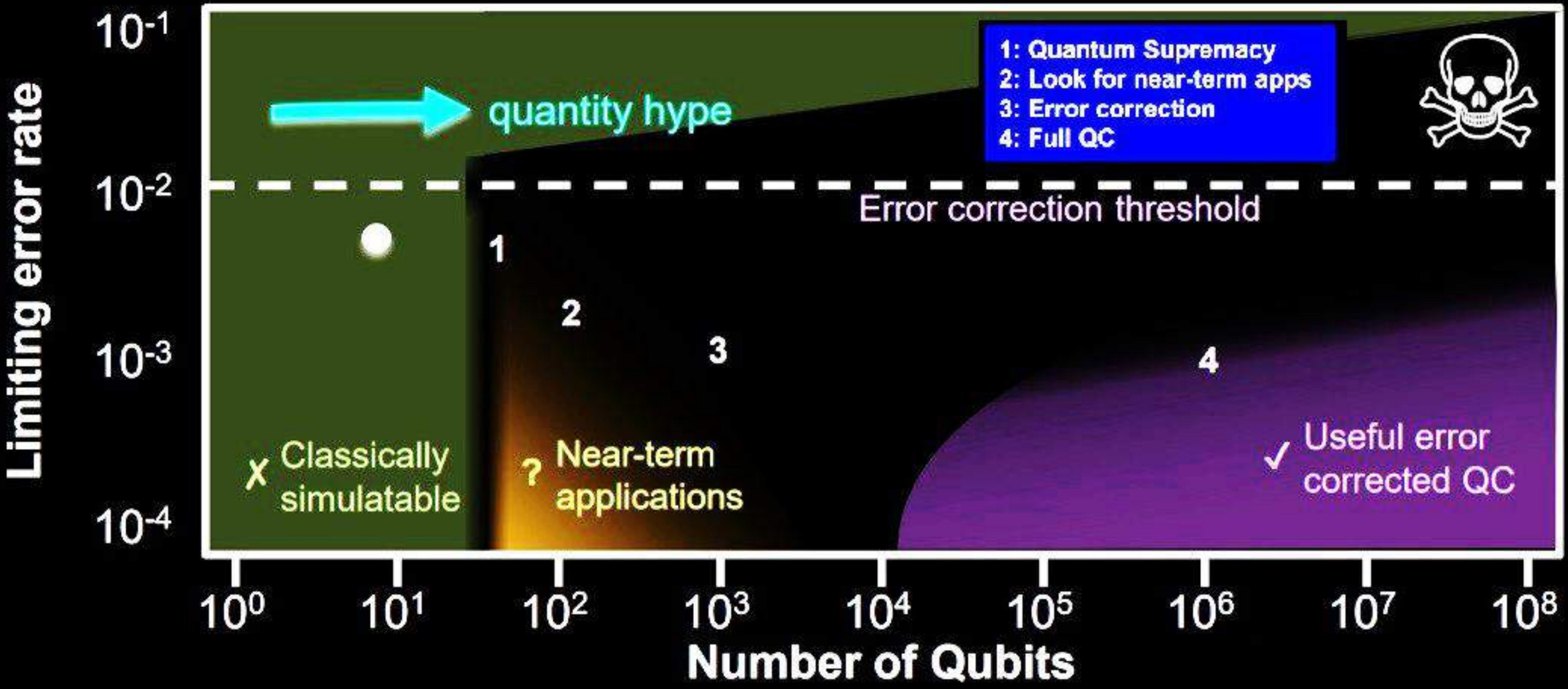


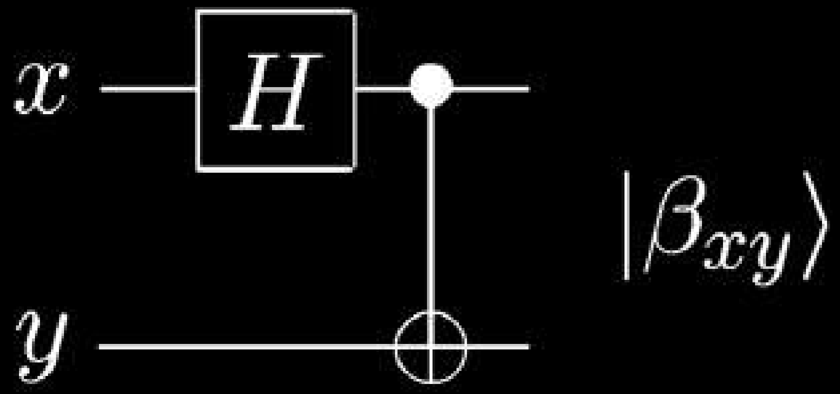
Illustration of the qubit quality vs quantity relationship. (Image credit: John Martinis, Google)

Quantum Computing

Quantum circuit to create Bell states, and its quantum ‘truth table’.*

$$|\beta_{xy}\rangle \equiv \frac{|0, y\rangle + (-1)^x |1, \bar{y}\rangle}{\sqrt{2}}$$

In	Out
$\|00\rangle$	$(\|00\rangle + \|11\rangle)/\sqrt{2} \equiv \|\beta_{00}\rangle$
$\|01\rangle$	$(\|01\rangle + \|10\rangle)/\sqrt{2} \equiv \|\beta_{01}\rangle$
$\|10\rangle$	$(\|00\rangle - \|11\rangle)/\sqrt{2} \equiv \|\beta_{10}\rangle$
$\|11\rangle$	$(\|01\rangle - \|10\rangle)/\sqrt{2} \equiv \|\beta_{11}\rangle$



*Nielsen & Chuang, *Quantum Computation and Quantum Information*, *10th-Anniversary Edition*, 2010, Cambridge University Press

Quantum Computing

Quantum circuit to create Bell state using Qiskit (IBM-Q 2 qubits):

Algorithm:

1. Initialize both qubits: $q_0 \leftarrow |0\rangle,\ q_1 \leftarrow |0\rangle$

2. Apply Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ on qubit q_0, which puts it into a **superposition** state:

$$q_0 \leftarrow Hq_0 = H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

3. Apply CNOT gate $C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$ on qubit q_0, using control qubit q_1, which puts both qubits in an **entangled** state:

$$|\psi\rangle = C_X(q_1 \otimes q_0) = C_X\left((|0\rangle) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\right) = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Quantum Computing

Quantum circuit to create Bell state using Qiskit (IBM-Q 2 qubits):

Algorithm (cont'd):

4. Measure the qubits:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

The probability of the measurement output:

$$P(\,|00\rangle\,) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}\,, \quad P(\,|01\rangle\,) = |0|^2 = 0\,, \quad P(\,|10\rangle\,) = |0|^2 = 0\,, \quad P(\,|11\rangle\,) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

5. Repeated many times measurements of the system will result in approximately 50% of $|00\rangle$ state and 50% of $|11\rangle$ state.

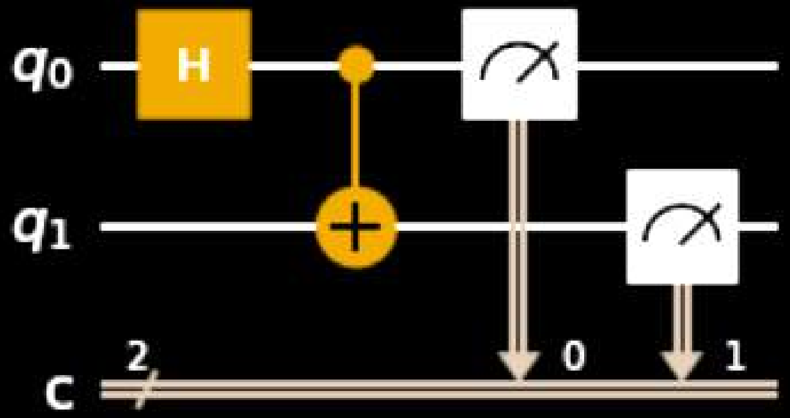
Quantum Computing: Let’s Coding!

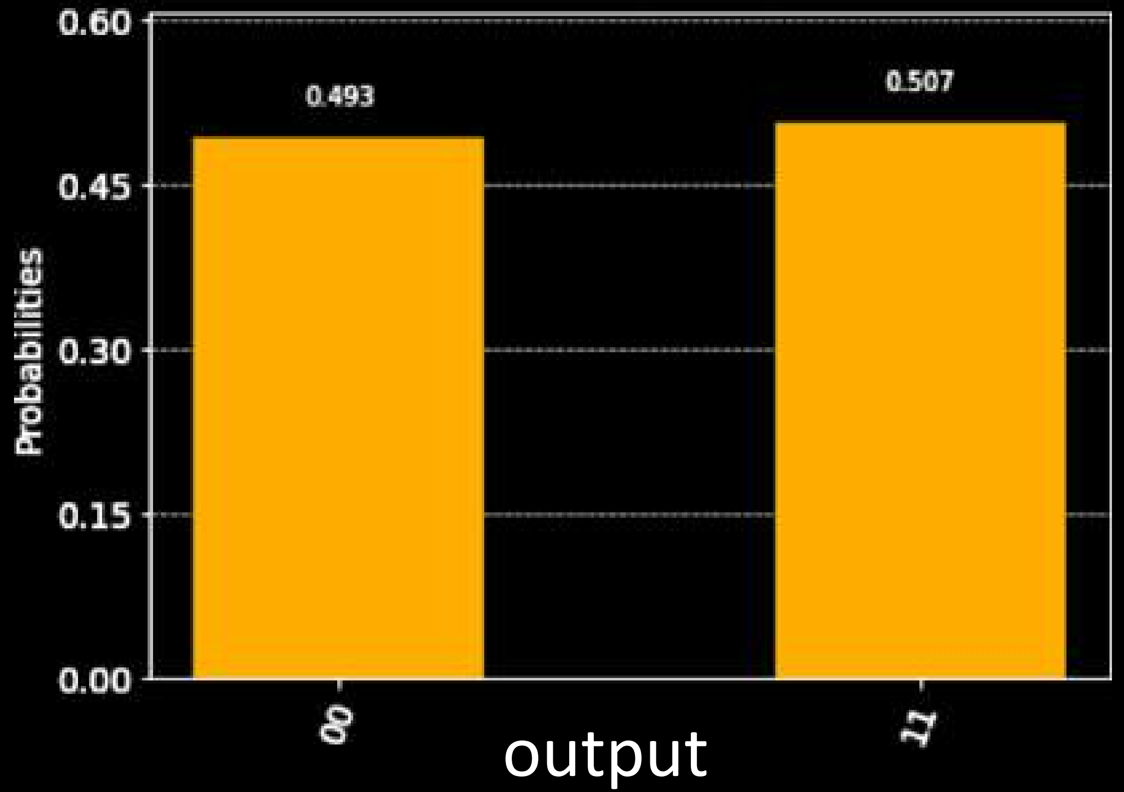
Hello Quantum World!   using Qiskit (IBM-Q 2 qubits):

Image credit: IBM

Quantum Computing

Quantum circuit to create Bell state using Qiskit (IBM-Q 2 qubits):

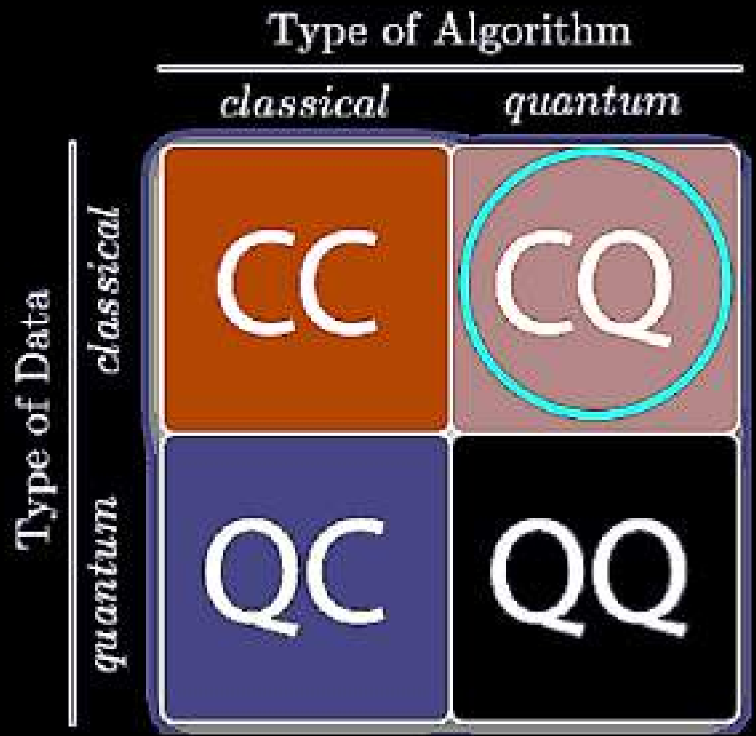




Quantum Artificial Intelligence

Quantum Machine Learning[1, 2]:

Using quantum computing resources for Machine Learning tasks



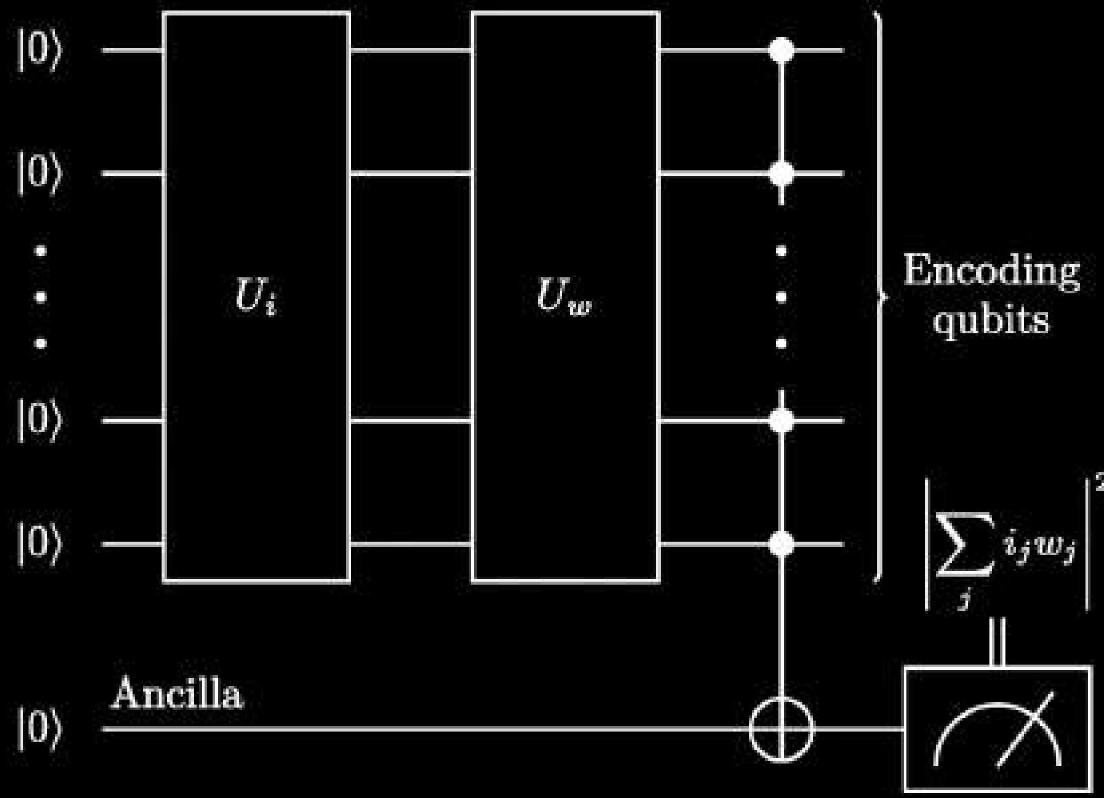
1. Biamonte, J., Wittek, P., Pancotti, N. et al. *Quantum machine learning*. Nature 549, 195–202 (2017)

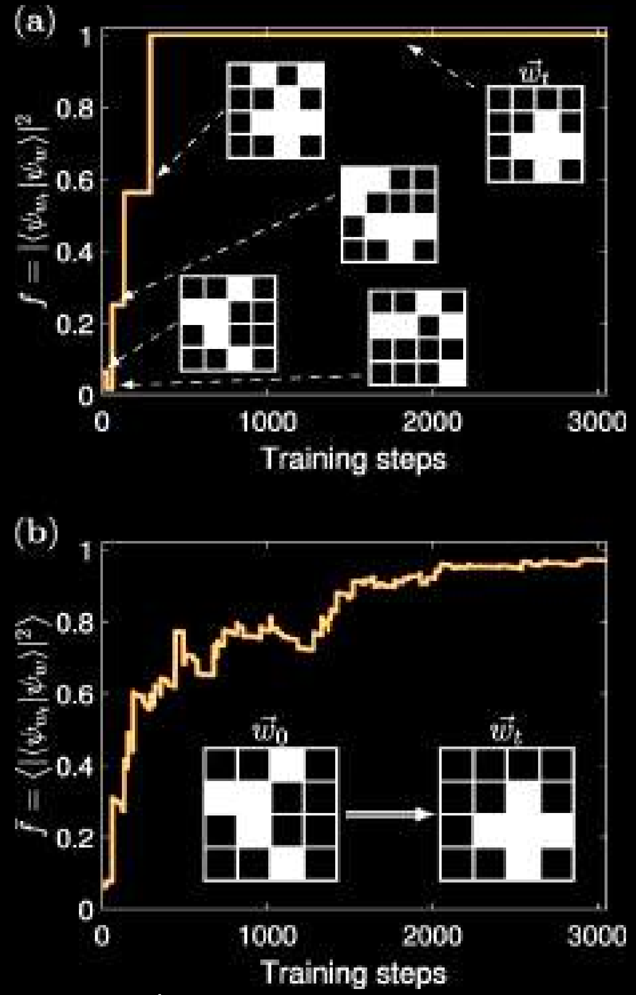
2. Maria Schuld, Francesco Petruccione, *Supervised Learning with Quantum Computers*, Springer (2018)

Quantum Artificial Intelligence: Example

An artificial neuron implemented on an actual quantum processor[1]:

Theoretical simulation of the algorithm for N = 4 qubits + 1 ancilla





Recognize a cross (or its negative) out of a training set of input vectors:

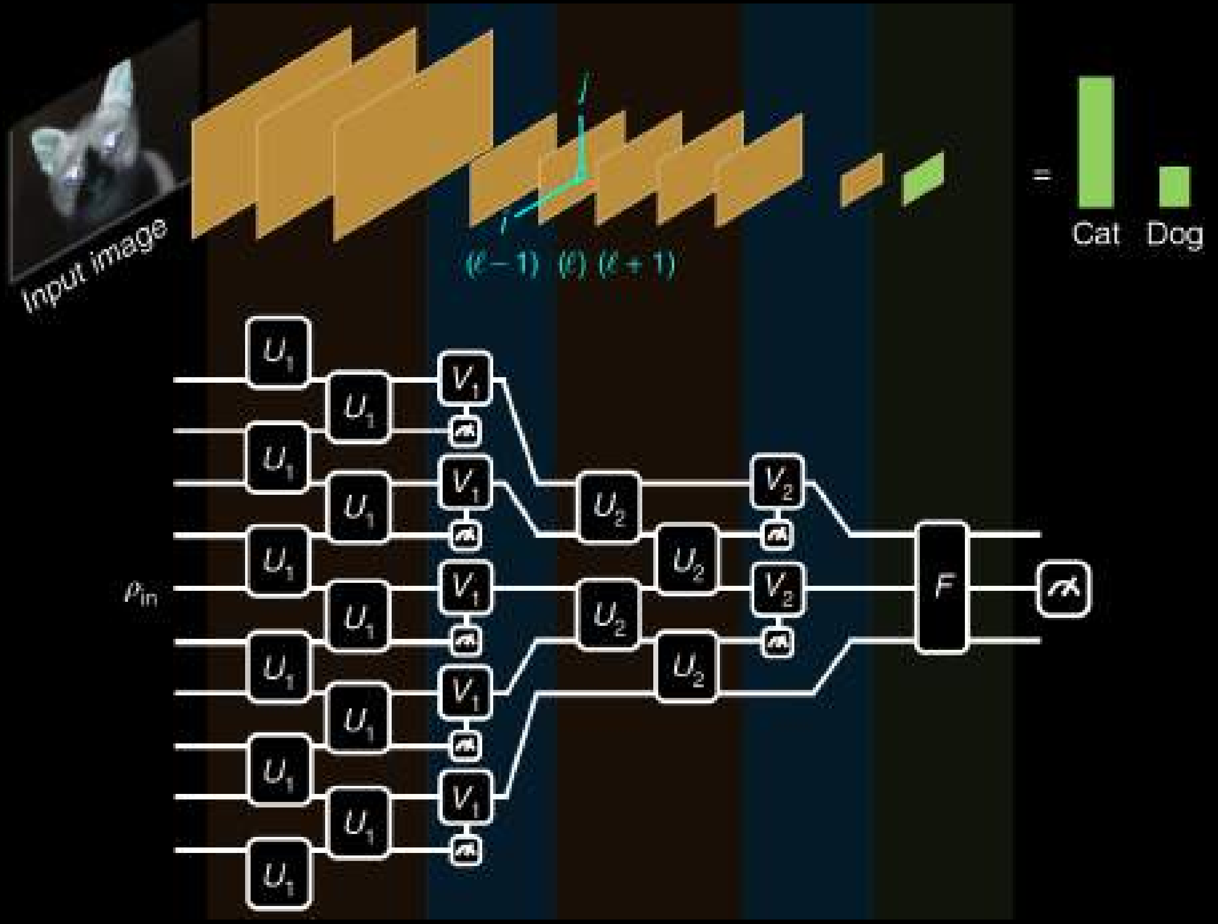
(a) A sample of training

(b) Average fidelity of the quantum state encoding the learned pattern with respect to the target one, obtained by repeating the learning procedure 500 times on the same training set

1. Tacchino, F., Macchiavello, C., Gerace, D. et al. *An artificial neuron implemented on an actual quantum processor*. npj Quantum Inf 5, 26 (2019)

Quantum Artificial Intelligence: Example

Quantum convolutional neural networks[1]**:**



A QCNN to classify N-qubit input states is thus characterized by $O(\log(N))$ parameters. This corresponds to a double exponential reduction compared with a generic quantum circuit-based classifier[19] and allows for efficient learning and implementation. For example, given a set of M classified training vectors $\{(|\psi_\alpha\rangle, y_\alpha): \alpha = 1, \ldots, M\}$, where $|\psi_\alpha\rangle$ are input states and $y_\alpha = 0$ or 1 are corresponding binary classification outputs, one could compute the mean squared error

$$\text{MSE} = \frac{1}{2M}\sum_{\alpha=1}^{M}\left(y_i - f_{\{U_i,V_j,F\}}(|\psi_\alpha\rangle)\right)^2 \quad (1)$$

Here, $f_{\{U_i,V_j,F\}}(|\psi_\alpha\rangle)$ denotes the expected QCNN output value for input $|\psi_\alpha\rangle$. Learning then consists of initializing all unitaries and successively optimizing them until convergence, for example via gradient descent.

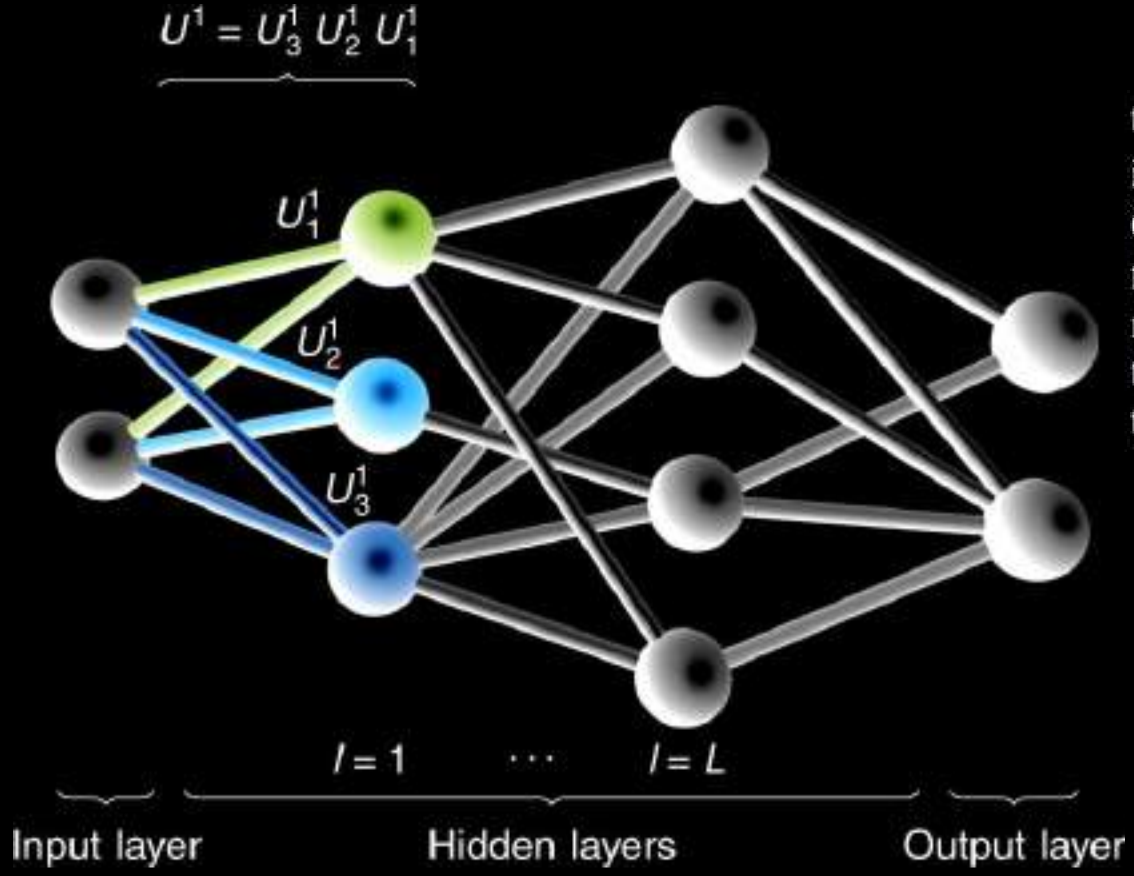
$O(\frac{7}{2}N(1-3^{1-d}) + 3^{1-d}N)$ multi-qubit operations

$O(4d)$ single-qubit rotations

1. Cong, I., Choi, S. & Lukin, M.D. *Quantum convolutional neural networks*. Nature Physics 15, 1273–1278 (2019)

Quantum Artificial Intelligence: Example

Training deep quantum neural networks[1]:



To evaluate the performance of our QNN in learning the training data, i.e., how close is the network output ρ_x^{out} for the input $|\phi_x^{\text{in}}\rangle$ to the correct output $|\phi_x^{\text{out}}\rangle$, we need a cost function. Operationally, there is an essentially unique measure of closeness for (pure) quantum states, namely the fidelity, and it is for this reason that we define our cost function to be the fidelity between the QNN output and the desired output averaged over the training data:

$$C = \frac{1}{N}\sum_{x=1}^{N} \langle \phi_x^{\text{out}} | \rho_x^{\text{out}} | \phi_x^{\text{out}} \rangle. \tag{3}$$

Maximizing cost function C

0 (worst) $\leq C \leq 1$ (best)

1. Beer, K., Bondarenko, D., Farrelly, T. et al. *Training deep quantum neural networks*. Nature Communications 11, 808 (2020)

Quantum Artificial Intelligence

Quantum **Speedup** for Unsupervised Learning*

Clustering problem	Classical	Quantum mechanical
Minimum spanning tree/k-clustering with maximum spacing	$\Theta(n^2)$	$\Theta(n^{3/2})$
Divisive clustering (in the case of balanced sub-clusters)	$\Theta(n^2)$	$\Omega(n)$, $O(n \log n)$
k-medians (standard)	$\Omega(\frac{n^2}{k})$, $O(t\frac{n^2}{k})$	$\Omega(n)$, $O(t\frac{1}{\sqrt{k}}n^{3/2})$
k-medians (distributed), communication cost	$\Theta(dn)$	$\Omega(d\sqrt{n})$, $O(t\sqrt{kn}(d + \log sum\_dist_{max}))$
Construction of a neighbourhood graph (for d a medium or high dimension)	$\Theta(n^2)$	$\Omega(n)$, $O(\sqrt{k}\,n^{3/2})$
Outlier detection (based on the neighbourhood graph)	$\Theta(n^2)$	$\Omega(n)$, $O(\sqrt{k}\,n^{3/2})$
"Smart" initialization of the cluster centres	$\Omega(n)$, $O(k^2 n)$	$\Omega(\sqrt{n})$, $O(k^2\sqrt{n})$

*Aïmeur, E., Brassard, G. & Gambs, S. *Quantum speed-up for unsupervised learning*. Mach Learn 90, 261–287 (2013).

Quantum Artificial Intelligence

Quantum **Speedup** for Some Supervised Learning[1, 2]

Method	Speedup = $\frac{O_{\text{Classical}}(\dots)}{O_{\text{Quantum}}(\dots)}$
Quantum inference on Bayesian Networks[3, 4]	$O(\sqrt{n})$
Online Perceptron[5]	$O(\sqrt{n})$
Quantum algorithm for Least Square Fitting[6]	$O(\log n)$
Classical Boltzmann machine using Gradient Estimation via Quantum Sampling (GEQS) & Gradient Estimation via Quantum Amplitude Estimation (GEQAE)[7]	$O(\sqrt{n})$
Quantum Boltzmann machine[8, 9]	$O(\log n)$
Quantum principal component analysis (PCA)[10]	$O(\log n)$
Quantum support vector machine (SVM)[11]	$O(\log n)$
Quantum reinforcement learning[12]	$O(\sqrt{n})$

Quantum Artificial Intelligence

Quantum **Speedup** for Some Supervised Learning[1, 2]

References:

1. Biamonte, J., Wittek, P., Pancotti, N. et al. *Quantum machine learning*. Nature 549, 195–202 (2017)
2. Rønnow, T. F. et al. *Defining and detecting quantum speedup*. Science 345, 420–424 (2014)
3. Low, G. H., Yoder, T. J. & Chuang, I. L. *Quantum inference on Bayesian networks*. Phys. Rev. A 89, 062315 (2014)
4. Wiebe, N. & Granade, C. *Can small quantum systems learn?* Preprint at https://arxiv.org/abs/1512.03145 (2015)
5. Wiebe, N., Kapoor, A. & Svore, K. M. Quantum perceptron models. Adv. Neural Inform. Process. Syst. 29, 3999–4007 (2016)
6. Wiebe, N., Braun, D. & Lloyd, S. *Quantum algorithm for data fitting*. Phys. Rev. Lett. 109, 050505 (2012)
7. Wiebe, N., Kapoor, A. & Svore, K. M. *Quantum deep learning*. Preprint at https://arxiv.org/abs/1412.3489 (2014)
8. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B. & Melko, R. *Quantum Boltzmann machine*. Preprint at https://arxiv.org/abs/arXiv:1601.02036 (2016)
9. Kieferova, M. & Wiebe, N. *Tomography and generative data modeling via quantum Boltzmann training*. Preprint at https://arxiv.org/abs/1612.05204 (2016)
10. Lloyd, S., Mohseni, M. & Rebentrost, P. *Quantum principal component analysis*. Nat. Phys. 10, 631–633 (2014)
11. Rebentrost, P., Mohseni, M. & Lloyd, S. *Quantum support vector machine for big data classification*. Phys. Rev. Lett. 113, 130503 (2014)
12. Dunjko, V., Taylor, J. M. & Briegel, H. J. *Quantum-enhanced machine learning*. Phys. Rev. Lett. 117, 130501 (2016)

Quantum Artificial Intelligence

Quantum Perceptrons:

Schuld et al., *Simulating a perceptron on a quantum computer*, Phys. Lett. A 7, 660-663 (2015)

N. Wiebe, A. Kapoor and K. M. Svore, *Quantum Perceptron Models*, arXiv:1602.04799 (2016)

Y. Cao, G. G. Guerreschi and A. Aspuru-Guzik, *Quantum Neuron: an elementary building block for machine learning on quantum computers*, arXiv:1711.11240 (2017)

Torrontegui et al., *Unitary quantum perceptron as efficient universal approximator*, EPL (Europhysics Letters) 125 (2019)

Tacchino, F., Macchiavello, C., Gerace, D. et al. *An artificial neuron implemented on an actual quantum processor*. npj Quantum Inf 5, 26 (2019)

Quantum Artificial Intelligence

Quantum artificial neural networks algorithms:

Beer, K., Bondarenko, D., Farrelly, T. et al. *Training deep quantum neural networks*. Nat Commun 11, 808 (2020)

Schuld et al., *Implementing a distance-based classifier with a quantum interference circuit*, EPL 119, 60002 (2017)

Wan, K.H., Dahlsten, O., Kristjánsson, H. et al. *Quantum generalisation of feedforward neural networks*. npj Quantum Inf 3, 36 (2017)

E. Farhi and H. Neven, *Classification with Quantum Neural Networks on Near Term Processors*, arXiv: 1802.06002 (2018)

Rebentrost et al., *Quantum Hopfield neural network*, Phys. Rev. A 98, 042308 (2018)

Grant et al., *Hierarchical quantum classifiers*, npj Quant Info 4, 65(2018)

Killoran et al., *Continuous-variable quantum neural networks*, Phys. Rev. Research 1, 033063 (2019)

Cong, I., Choi, S. & Lukin, M.D. *Quantum convolutional neural networks*. Nature Physics 15, 1273–1278 (2019)

Mari et al., *Transfer learning in hybrid classical-quantum neural networks*, Quantum 4, 340 (2020)

Roadmap of Google Quantum Computing

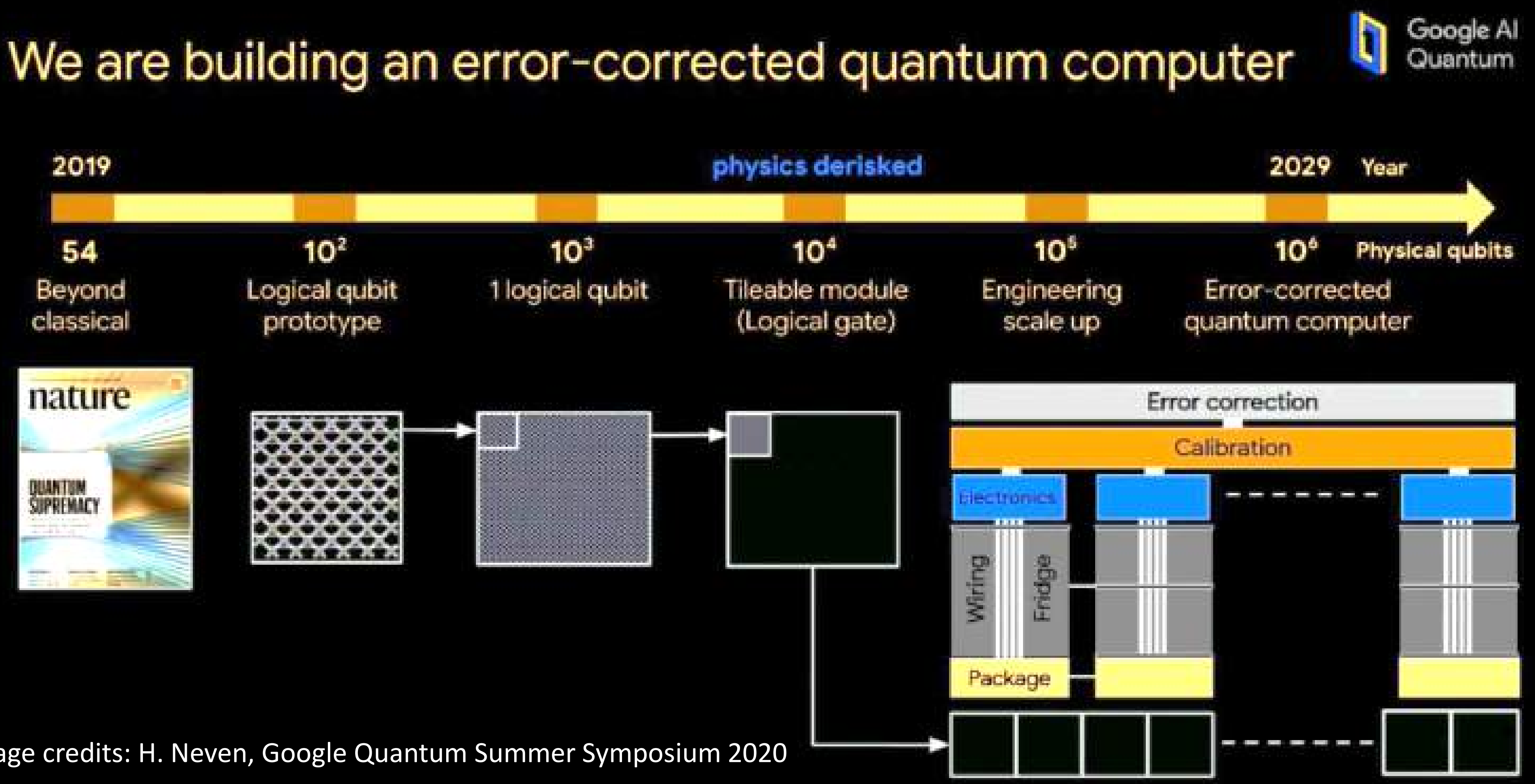


Image credits: H. Neven, Google Quantum Summer Symposium 2020

Roadmap of Honeywell Quantum Computing

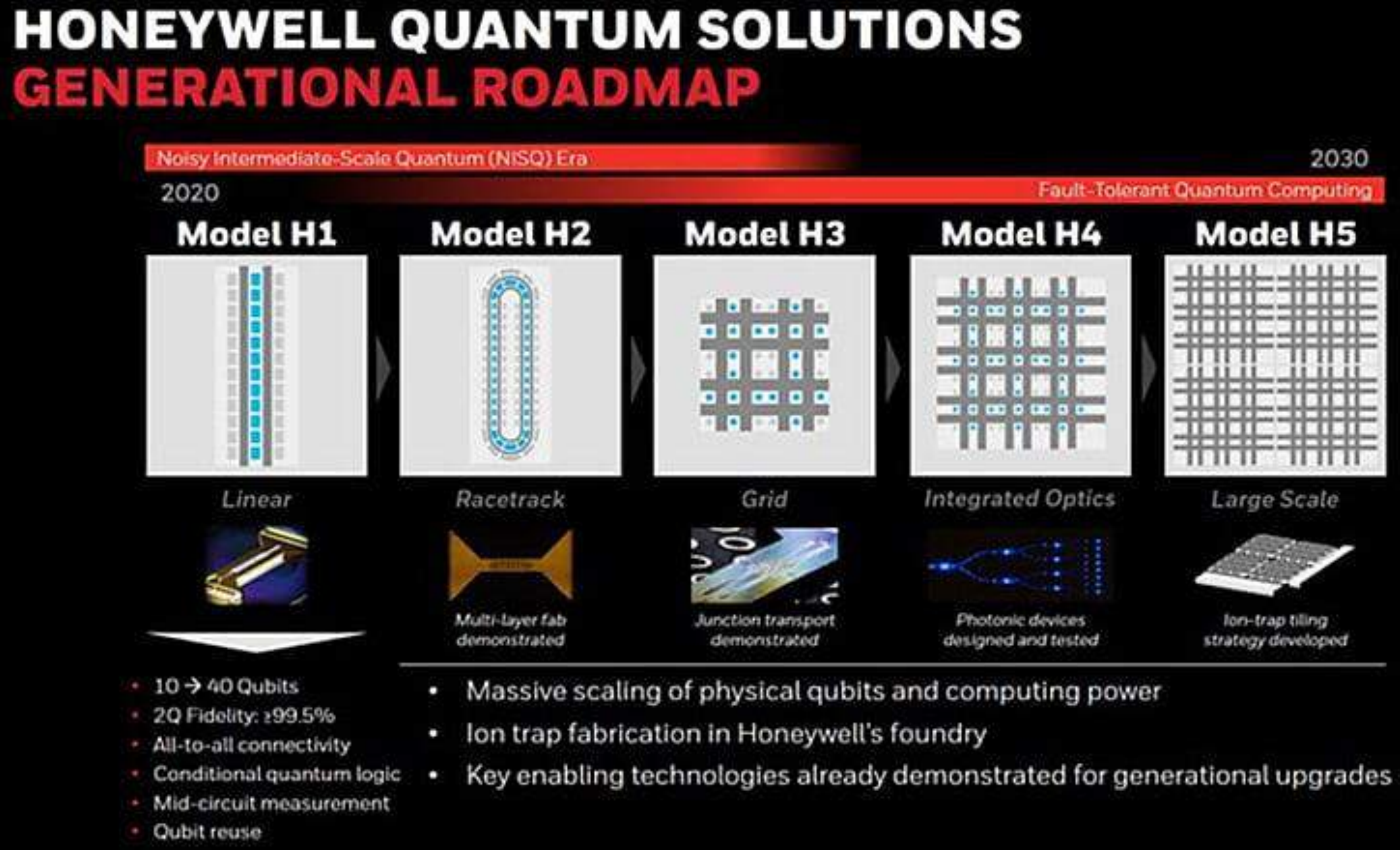


Image credits: Honeywell news post

Roadmap of IBM’s Scaling Quantum Technology

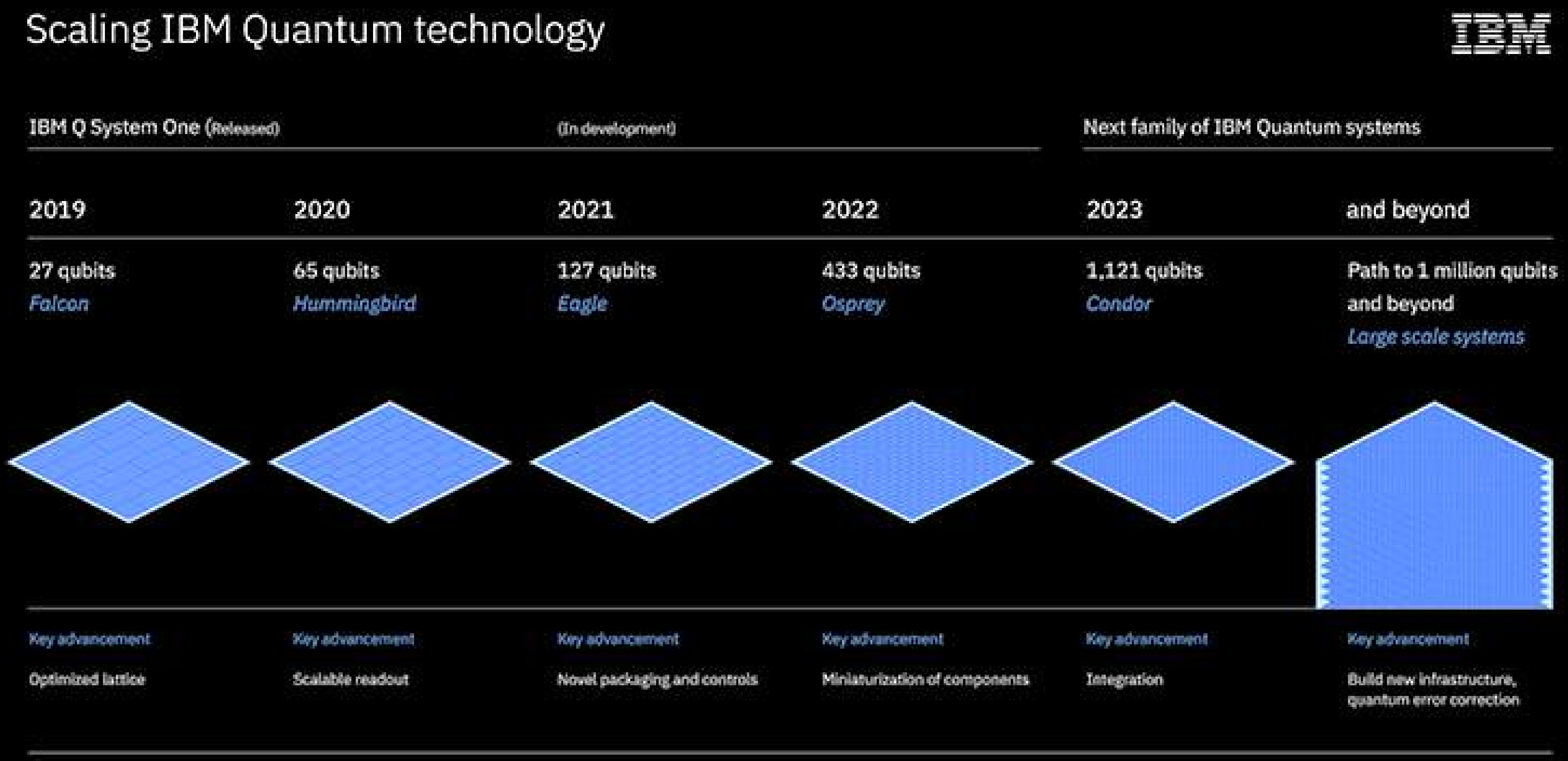


Image credits: https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap

Roadmap of Scaling IonQ's Quantum Computers

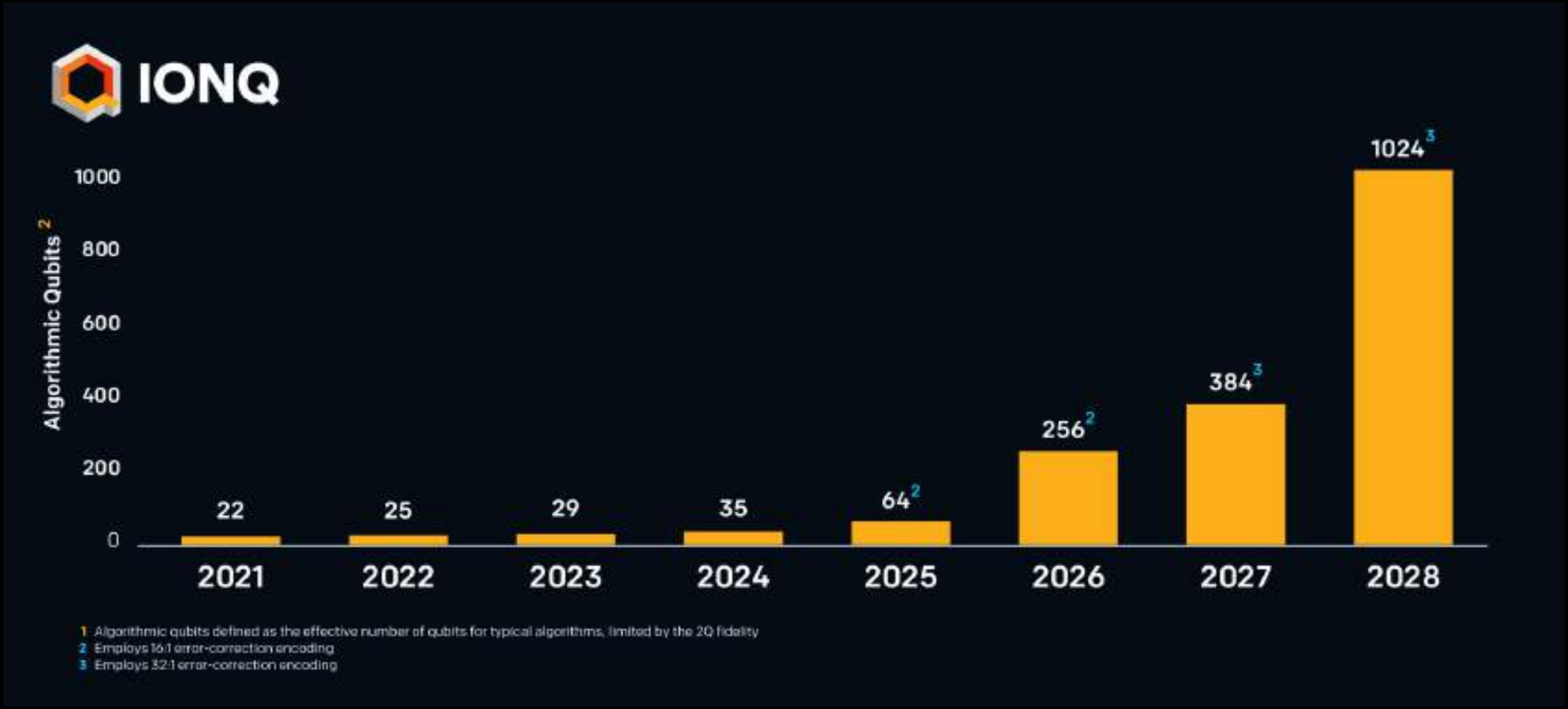


Image credits: https://ionq.com/posts/december-09-2020-scaling-quantum-computer-roadmap

Questions?

SELAMAT DATANG
DI UNIVERSITAS KRISTEN MARANATHA

THE FACULTY OF

INFORMATION TECHNOLOGY

NO LIMITS, NO BOUNDARIES

Find us

THE FACULTY OF
INFORMATION
TECHNOLOGY
NO LIMITS, NO BOUNDARIES

AYO KULIAH MAGISTER ILMU KOMPUTER BERSAMA FAKULTAS IT MARANATHA!

POTONGAN 2,5 JUTA/SEMESTER!

KULIAH DI FAKULTAS IT MARANATHA BISA LANGSUNG DAPAT 2 GELAR LHO!!!