# Machine Learning

## Lesson 3

School on Data Science in Fundamental Physics, IGFAE/USC, Spain

Dr. Pietro Vischia
pietro.vischia@cern.ch
@pietrovischia
https://vischia.github.io/

Universidad de Oviedo

ICTEA

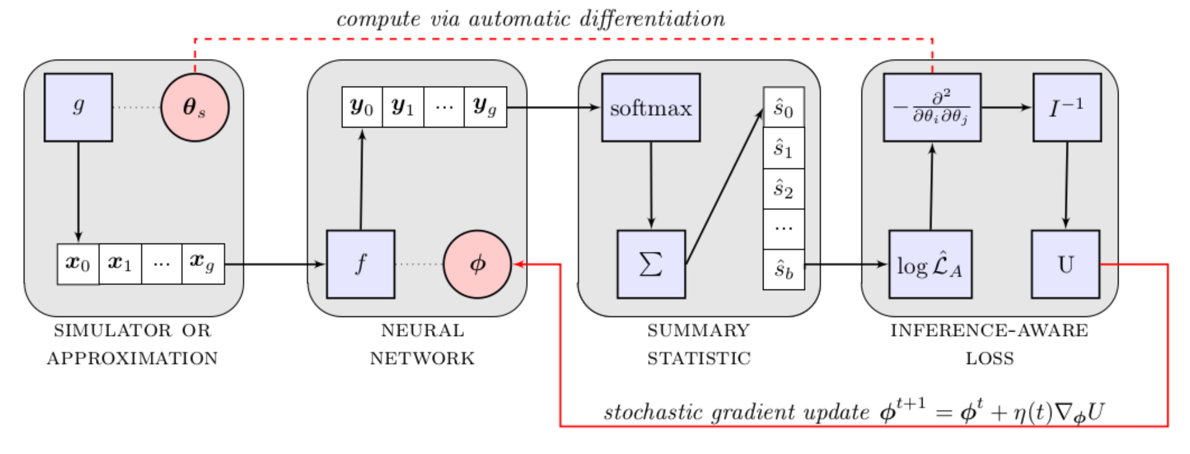
Supported by project
RYC2021- 033305-I
funded by
GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN
AGENCIA ESTATAL DE INVESTIGACIÓN

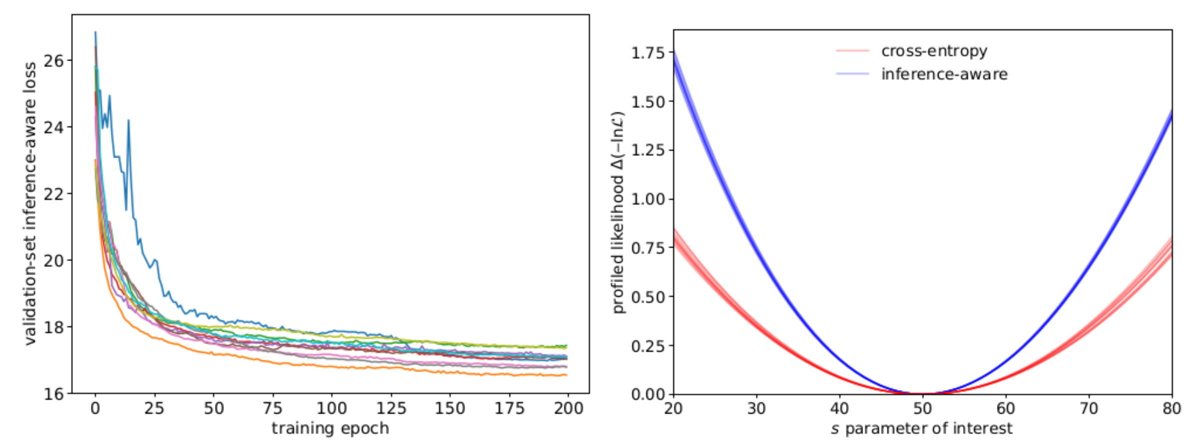
If you are reading this as a web page: have fun! If you are reading this as a PDF: please visit

https://www.hep.uniovi.es/vischia/persistent/2024-06-03to07_MachineLearningAtDataScienceSchoolIGFAE_vischia_3.html

to get the version with working animations

# Go to INFERNO: syst-aware inference opt.




(a) inference-aware training loss

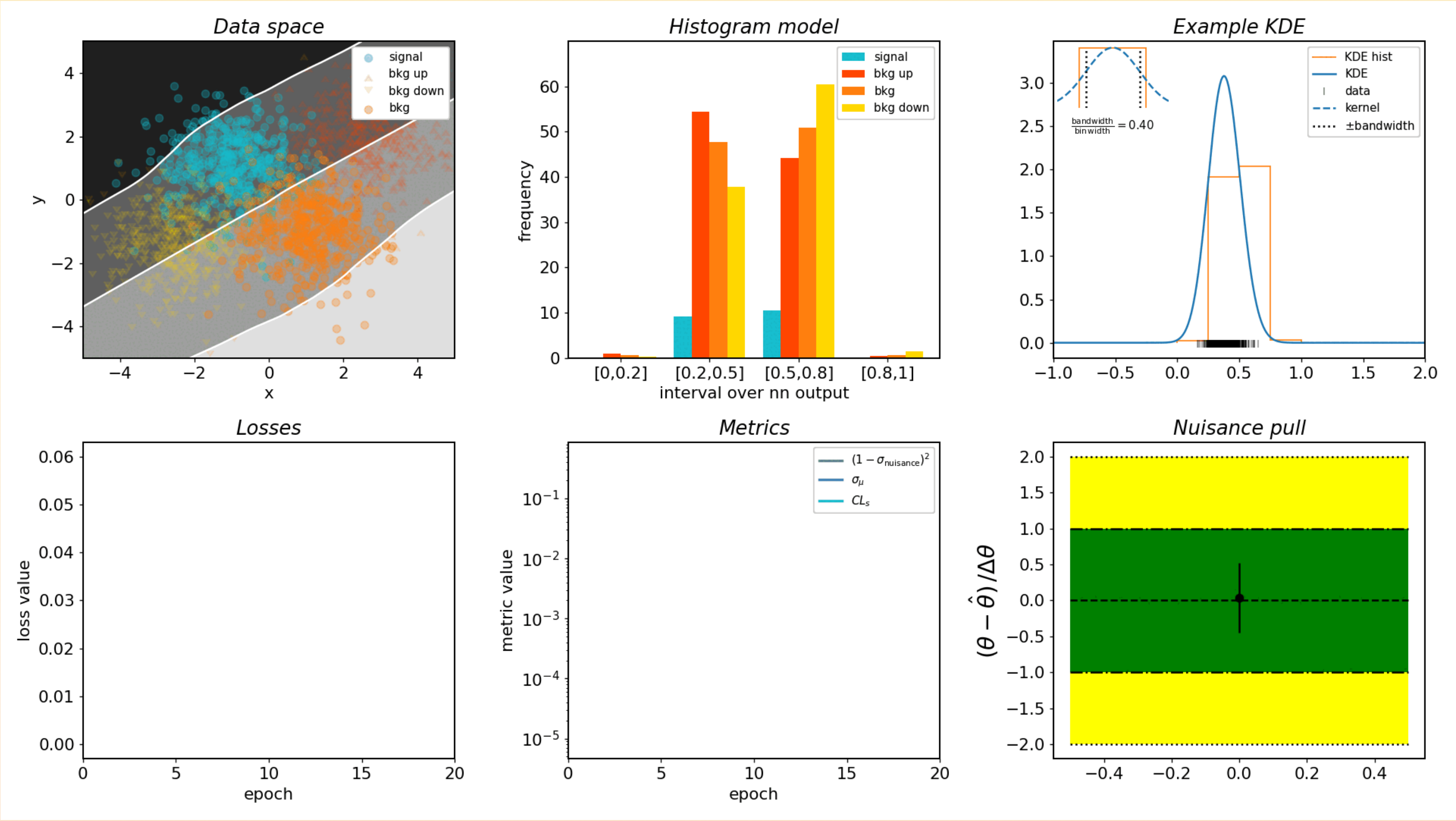
(b) profile-likelihood comparison

—

Figures from 10.1016/j.cpc.2019.06.007

# Measurement-aware analysis opt.

neos



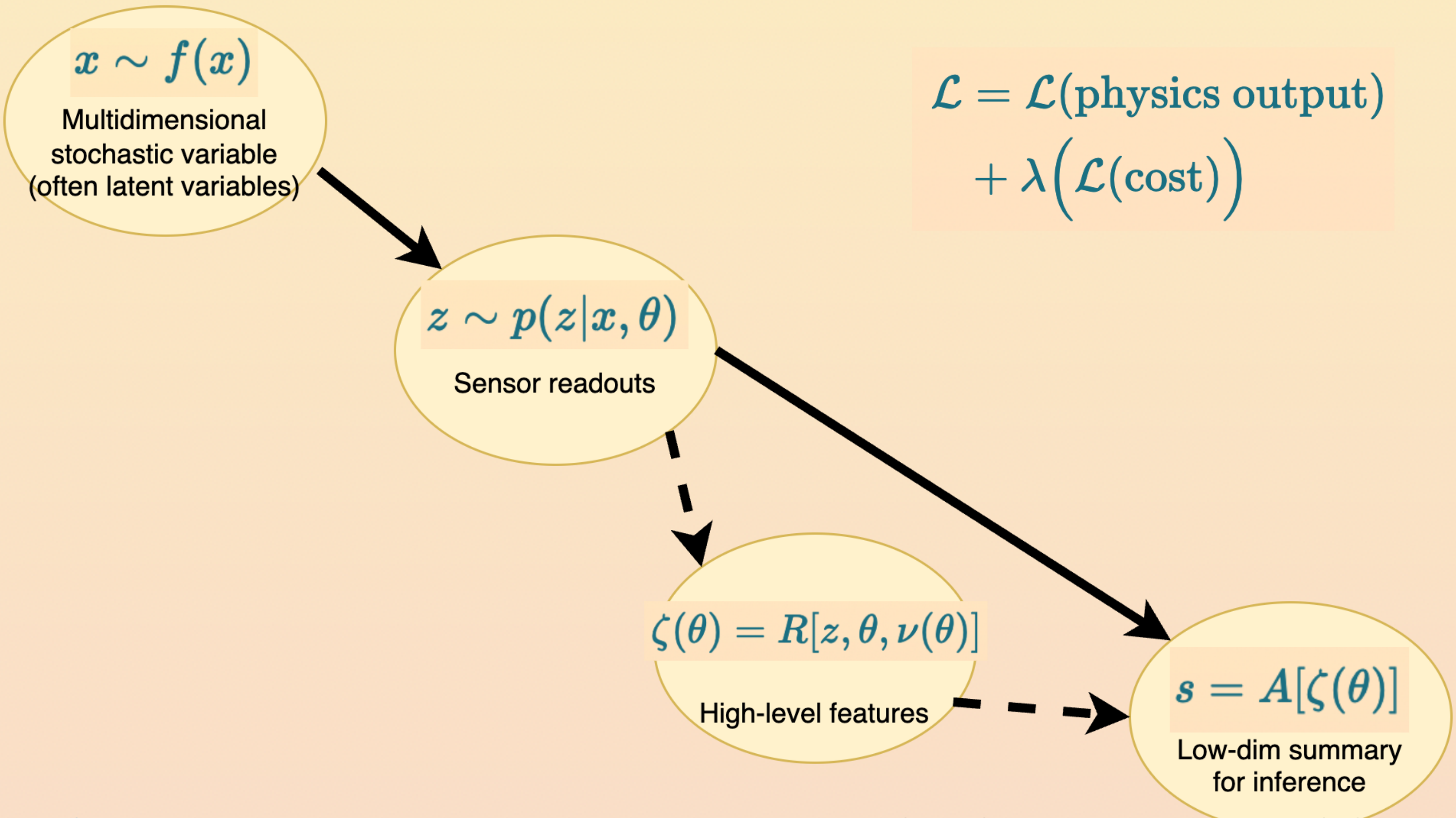

—

Animation from 2203.05570

# Measurement-aware detector opt.!

- Joint optimization of design parameters w.r.t. inference made with data
- MODE White Paper, 10.1016/j.revip.2023.100085 (2203.13818), 117-pages document, physicists + computer scientists (published on June 2023!!!)

$$\mathcal{L} = \mathcal{L}(\text{physics output}) + \lambda\Big(\mathcal{L}(\text{cost})\Big)$$




—
Formulas from our white paper

# Guarantee feasibility within constraints

- Monetary cost
- Case-specific technical constraints

$$\mathcal{L}_{\text{cost}} = c(\theta, \phi)$$

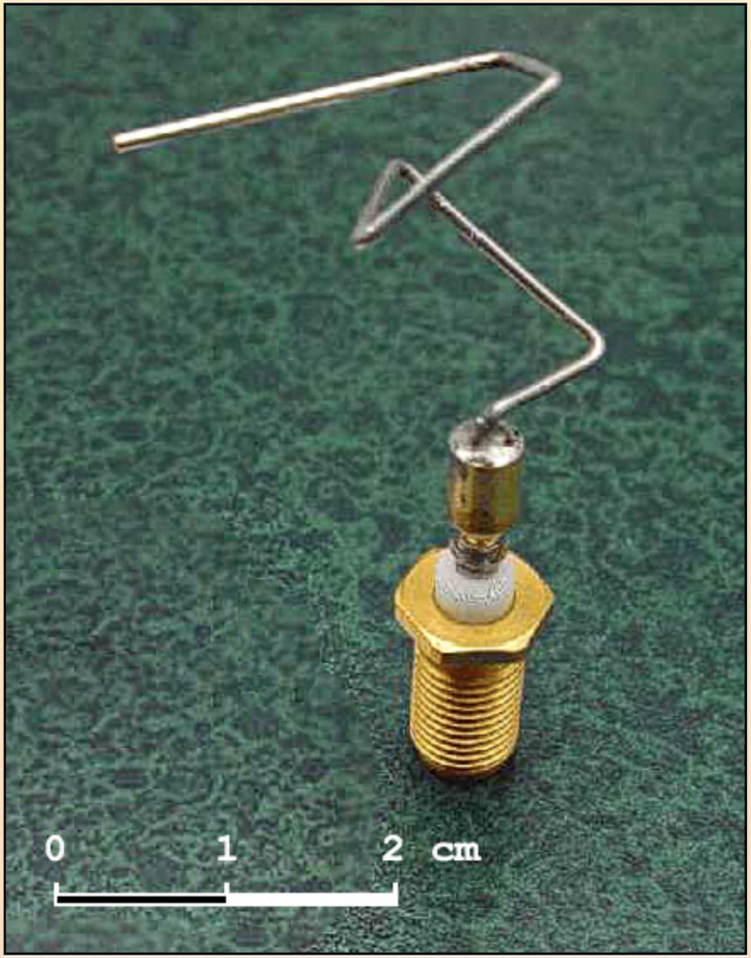
- $\theta$: local, specific to the technology used (e.g. active components material)
- $\phi$: global, describing overall detector conception (e.g. number, size, position of detector modules)
- Fixed costs can be added separately to the loss function

# In general

Depends on z and nuisances

$$\hat{\theta} = arg\ min_{\theta} \int \underbrace{L[A(\zeta), \overbrace{c(\theta)}^{\text{Cost of the layout with parameters theta}}]}_{\text{Weight desirable goals while obeying cost constraints}} \overbrace{p(z|x, \theta)}^{\text{Closed form}} f(x) dxdz\ ,$$

# Symmetry and interpretability




—

Photo from doi:10.2514/6.2006-7242

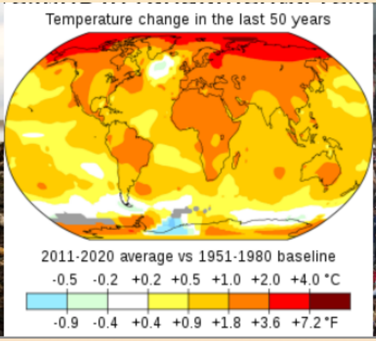
# Moral imperative

Optimize...

- New large, long-term projects
- Push technological skills to the limit (cit. EUSUPP)

...within constraints

- Unprecedented global challenges
- Society less receptive to fundamental research




Maximum extraction of scientific value from the available resources

—

2020 update of the European Strategy for Particle Physics (EUSUPP)

# The MODE Collaboration

https://mode-collaboration.github.io/

- Joint effort (created 11.2020) of particle physicists, nuclear physicists, astrophysicists, and computer scientists. IGFAE too, via Xabier Cid Vidal!

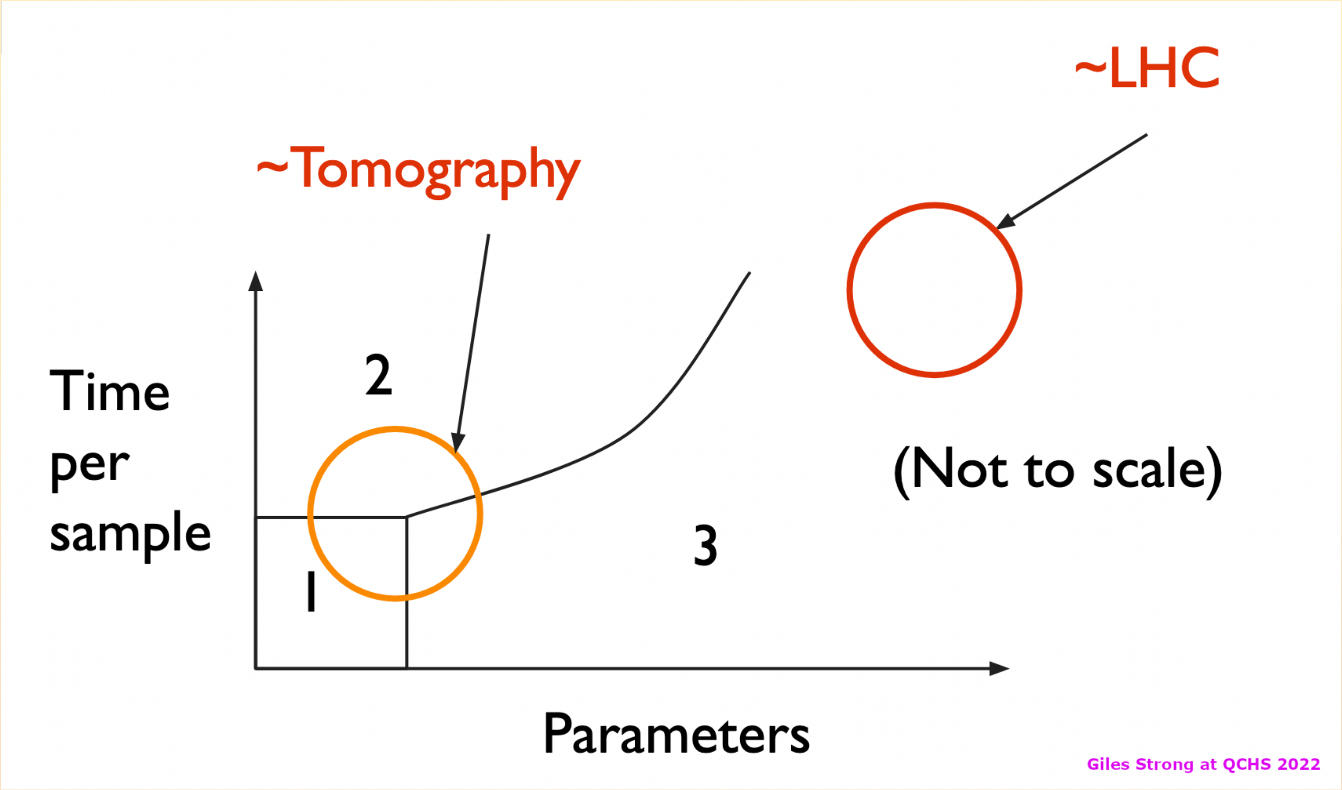
COLLABORATION

At INFN and Università of Padova Dr. **Tommaso Dorigo**, Dr. **Pablo De Castro Manzano**, Dr. **Federica Fanzago**, Dr. **Lukas Layer**, Dr. **Giles Strong**, Dr. **Mia Tosi**, and Dr. **Hevjin Yarar**
At Université catholique de Louvain Dr. **Andrea Giammanco**, Prof. **Christophe Delaere**, and Mr. **Maxime Lagrange**
At Universidad de Oviedo and ICTEA Dr. **Pietro Vischia**
At Université Clermont Auvergne, Prof. **Julien Donini**, and Mr. **Federico Nardi** (joint with Universitá di Padova)
At the Higher School of Economics of Moscow, Prof. **Andrey Ustyuzhanin**, Dr. **Alexey Boldyrev**, Dr. **Denis Derkach**, and Dr. **Fedor Ratnikov**
At the Instituto de Física de Cantabria, Dr. **Pablo Martínez Ruíz del Árbol**
At CERN, Dr. **Jan Kieseler**, Dr. **Sofia Vallecorsa**
At University of Oxford Dr. **Atilim Gunes Baydin**
At New York University Prof. **Kyle Cranmer**
At Université de Liège Prof. **Gilles Louppe**
At GSI/FAIR Dr. **Anastasios Belias**
At Rutgers University Dr. **Claudius Krause**
At Uppsala Universitet Prof. **Christian Glaser**
At TU-München, Prof. **Lukas Heinrich** and Mr. **Max Lamparth**
At Durham University Dr. **Patrick Stowell**
At Lebanese University Prof. **Haitham Zaraket**
At University of Kaiserslautern-Landau Mr. **Max Aehle**, Prof. **Nicolas Gauger**, Dr. **Lisa Kusch**
At University of Applied Sciences Worms Prof. **Ralf Keidel**
At Princeton University Prof. **Peter Elmer**
At University of Washington Prof. **Gordon Watts**
At SLAC Dr. **Ryan Roussel**
At Lulea University of Technology Prof. **Fredrik Sandin** and Prof. **Marcus Liwicki**
At IGFAE and Universidad de Santiago de Compostela Prof. **Xabier Cid Vidal**

# MODE Workshop Series

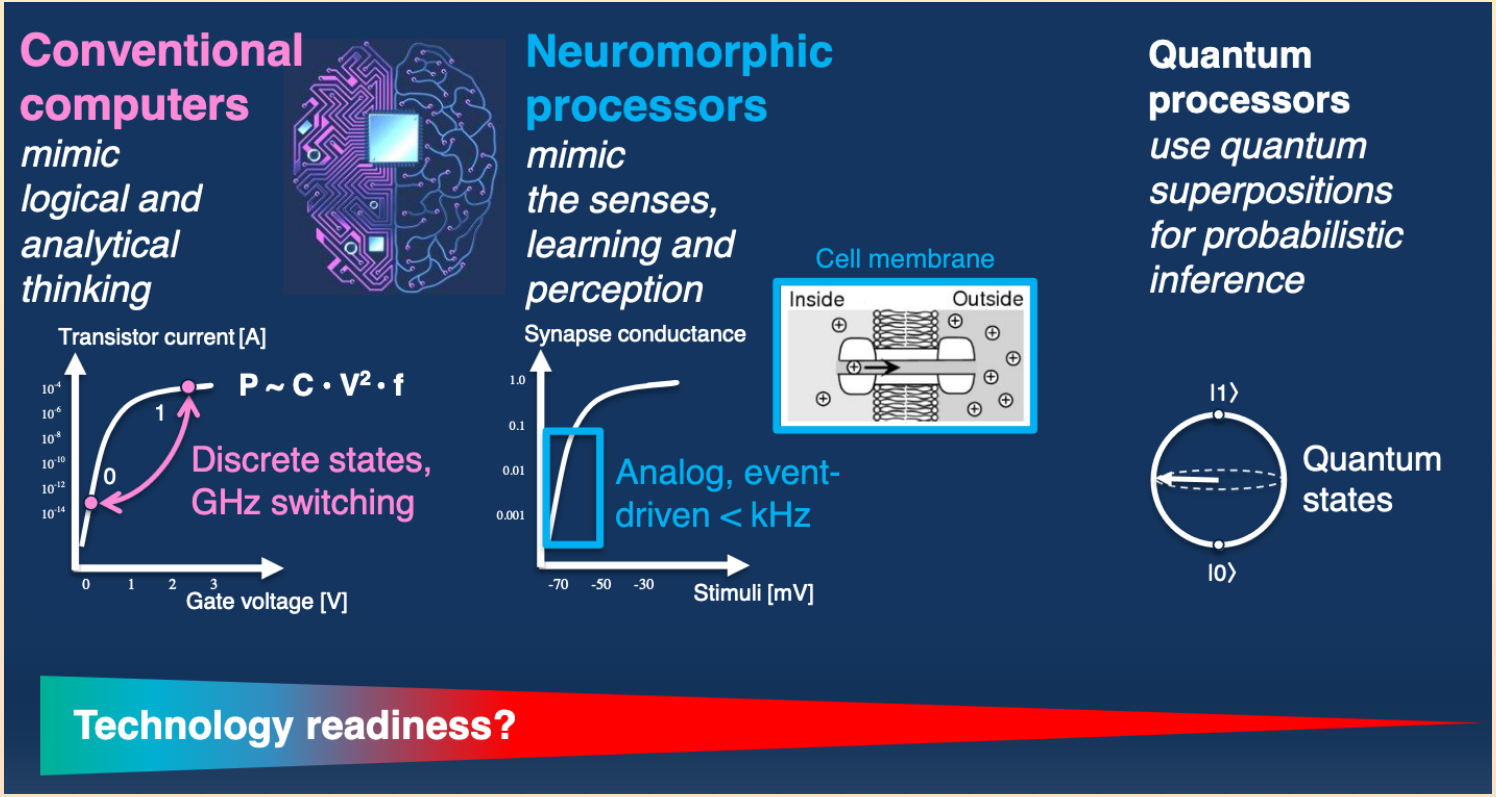
- Yearly workshops on Differentiable Programming for Experiment Design
  - 2021: First Edition (Louvain-la-Neuve, Belgium)
  - 2022: Second Edition (Kolymbari, Greece)
  - 2023: Third Edition (Princeton, USA)
- You are all invited to the Fourth Workshop, to be held in Valencia (Spain), 23-25 September 2024!!!
  - Abstract submission is still open!

# Method of choice depends on scale




1. Grid/random search
2. Bayesian opt, simulated annealing, genetic algos, ...
3. Gradient-based optimization (Newton, BFGS, gradient descent, ...)

# Need for new paradigma




—
Image by Fredrik Sandin, Lulea University

# Improve digital hardware

Computational/architectural tricks, or fast chips (FPGA, ASIC)

# Energy efficiency: redundancy removal

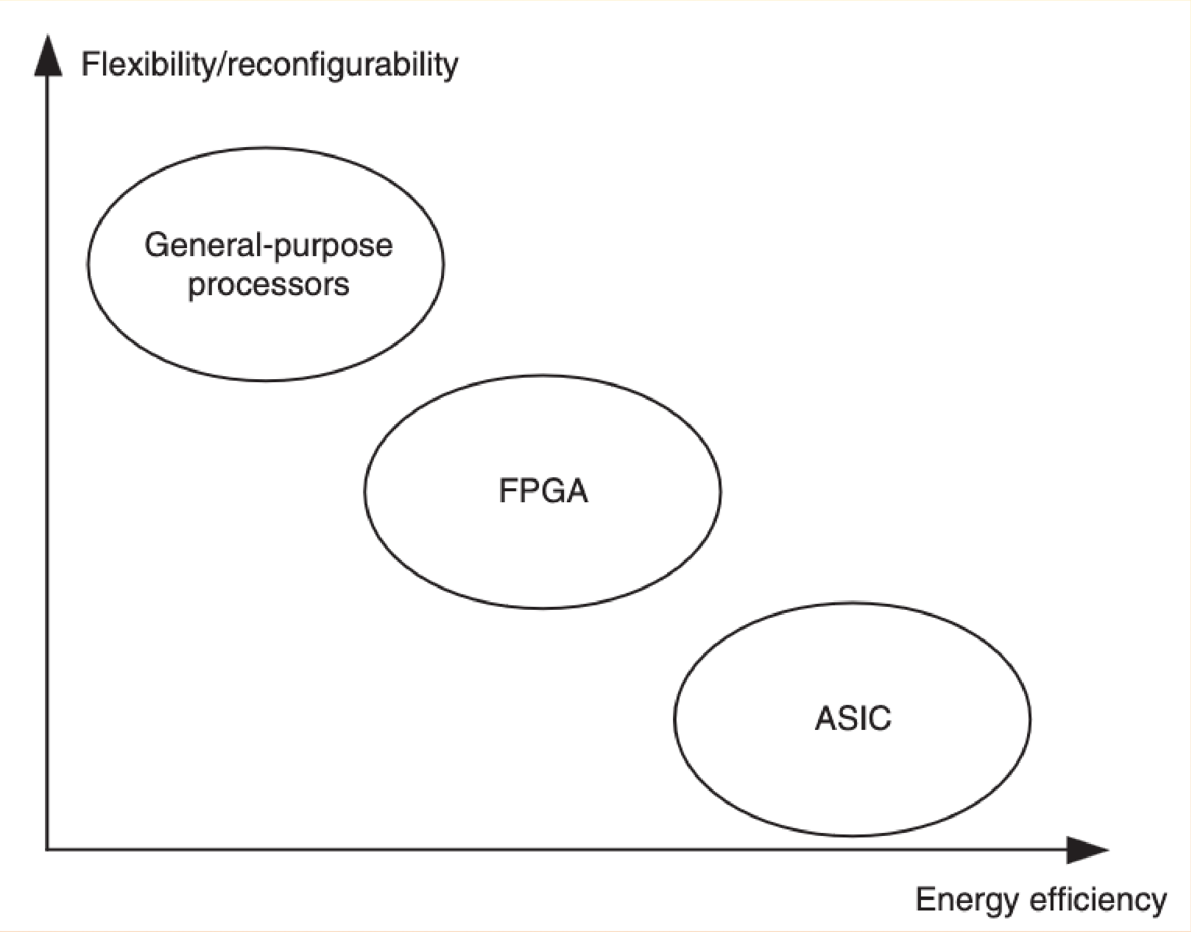
- Synaptic weights storage and operation on them (e.g. memory access) bulk of energy-consuming operations
- Less weights $\longrightarrow$ less energy and time consumption
  - Weight pruning, low-rank approximations, etc
- Example: Yann LeCun's Optimal Brain Damage Figure 2.26mazumder
- Sparse activation patterns (via gating)

# Energy efficiency: precision reduction

- Quantization

https://fastmachinelearning.org/hls4ml/

# Neural Networks in hardware




—
Image from Zheng, Mazumder (Wiley, 2019)

# Data Movements cost

- CPU design: dataflow hard to predict
- Neural network accelerators: dataflow often fixed and known at compilation time

Can optimize data movement and memory access!

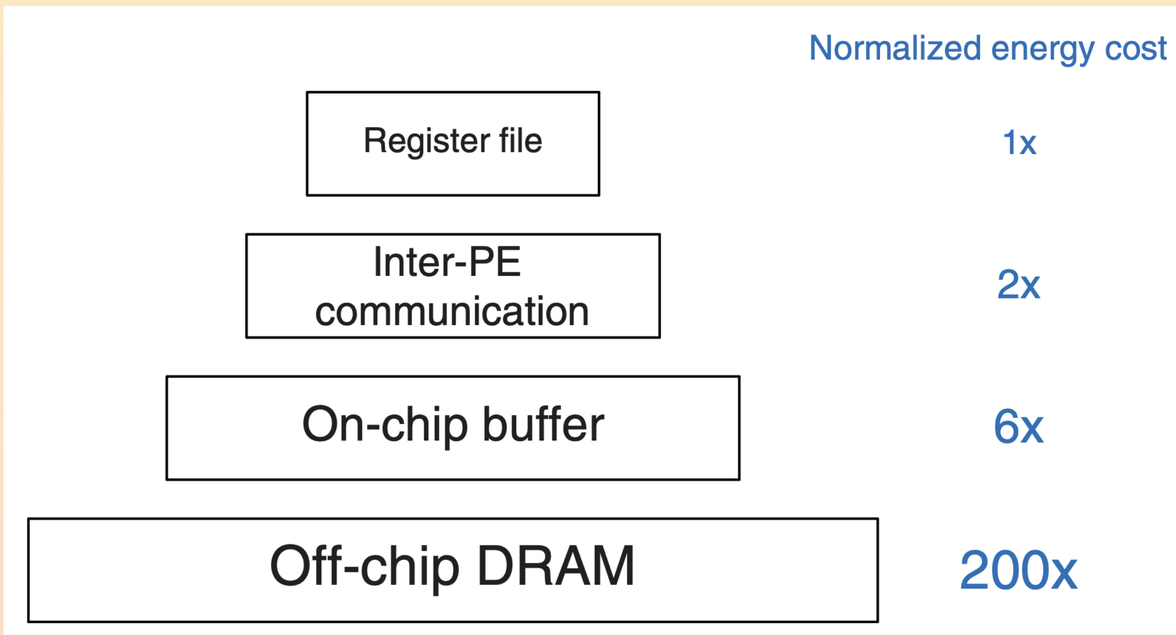
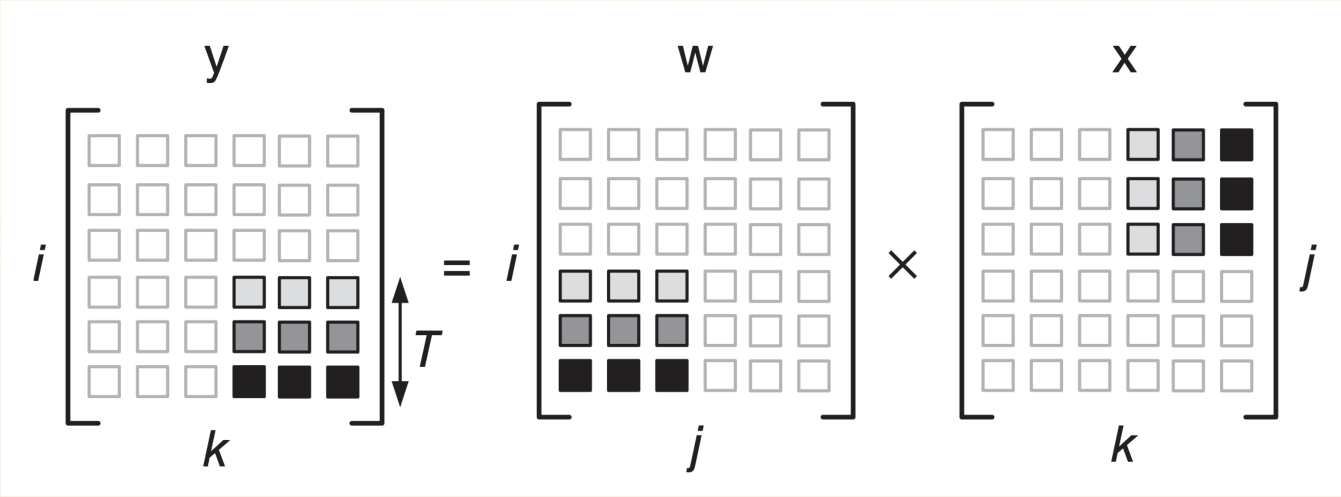



Image from Zheng, Mazumder (Wiley, 2019). It assumes that every processing element has a dedicated register file

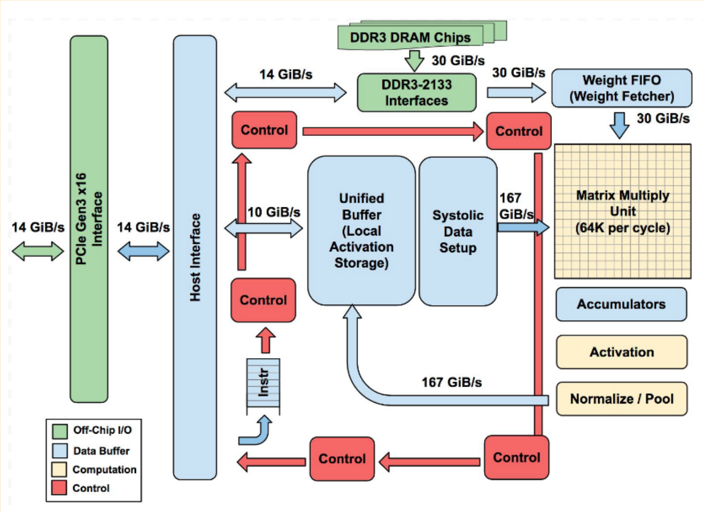
# Example: Tiling

- Break down matrix multiplication into subproblems that fit on-chip buffer
    - Maximise data reuse



# Example: Google's TPUs

- Systolic flow
  - Hide four-stage process within the matrix multiplication operation
  - E.g. decoupled access/execution when reading weights
  - Trick flow control into thinking inputs are read and update results at once



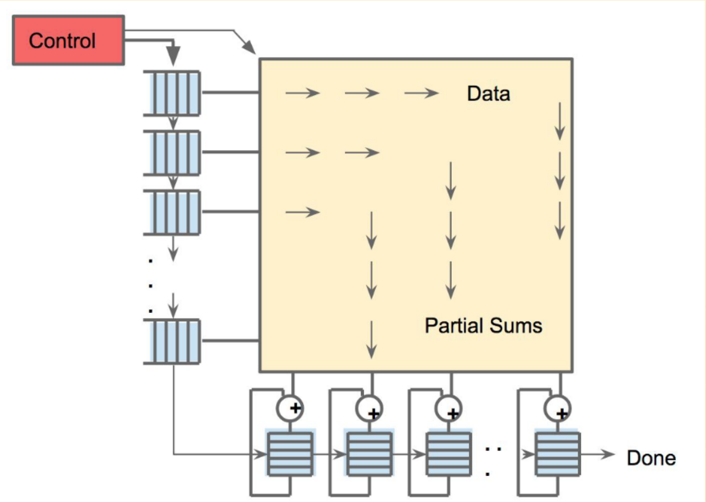




**Figure 4.** Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

# Example: FPGA

- More configurable than ASIC (but it consumes more)
- Covered by the other speakers!

# Neuromorphic computing

Go back to spike-based neural models

# Gymnotus Omarorum

- Lives in ponds, active mostly by night
  - Murky waters, lots of vegetation
- Generates electrical field, and detects its deformations
  - Receptors are relatively simple neurons
- Often used as bait for fishing other fishes (pirayú and surubí)

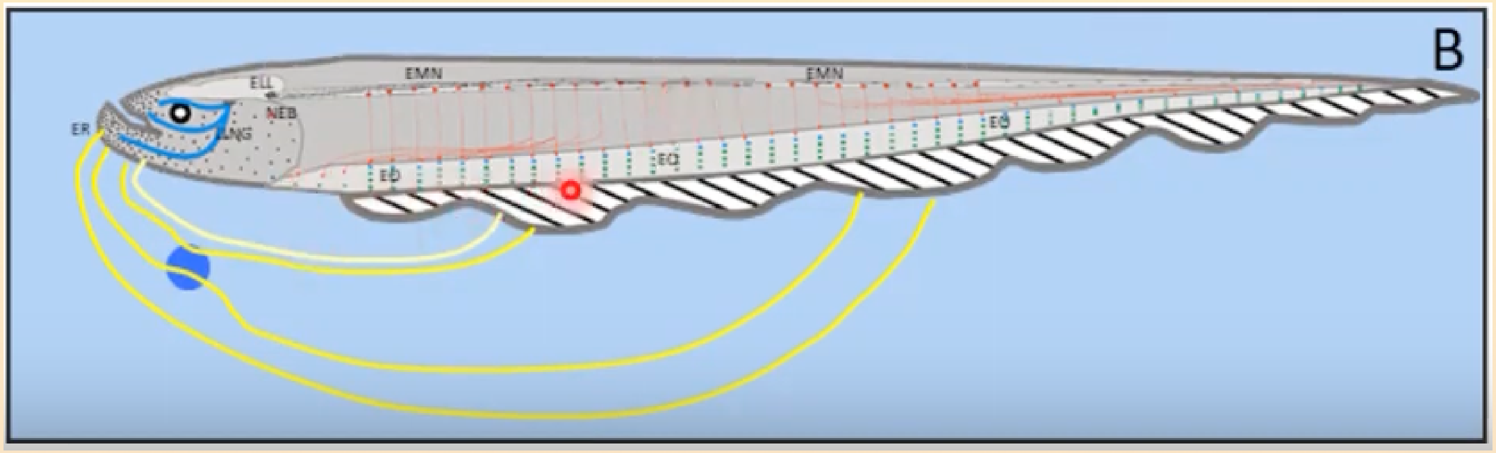
- Up to about 25 cm long

—

Image from Wikipedia

# Generation and Detection

- Electrical organs generate electrical field around the fish
- Sensorial Electroreceptors Organs detect the field and its changes



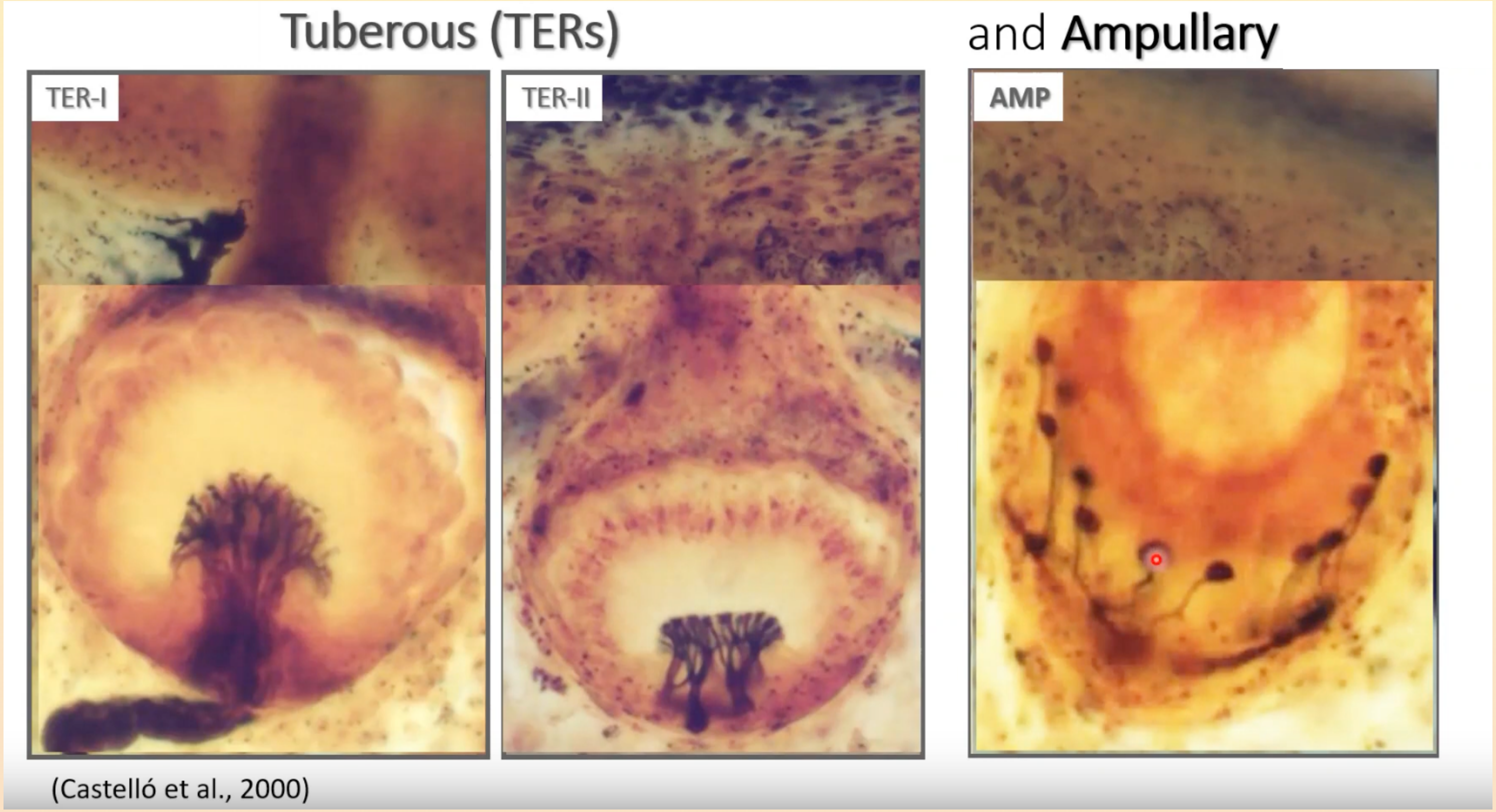

—

Image from Palave et al., 2022

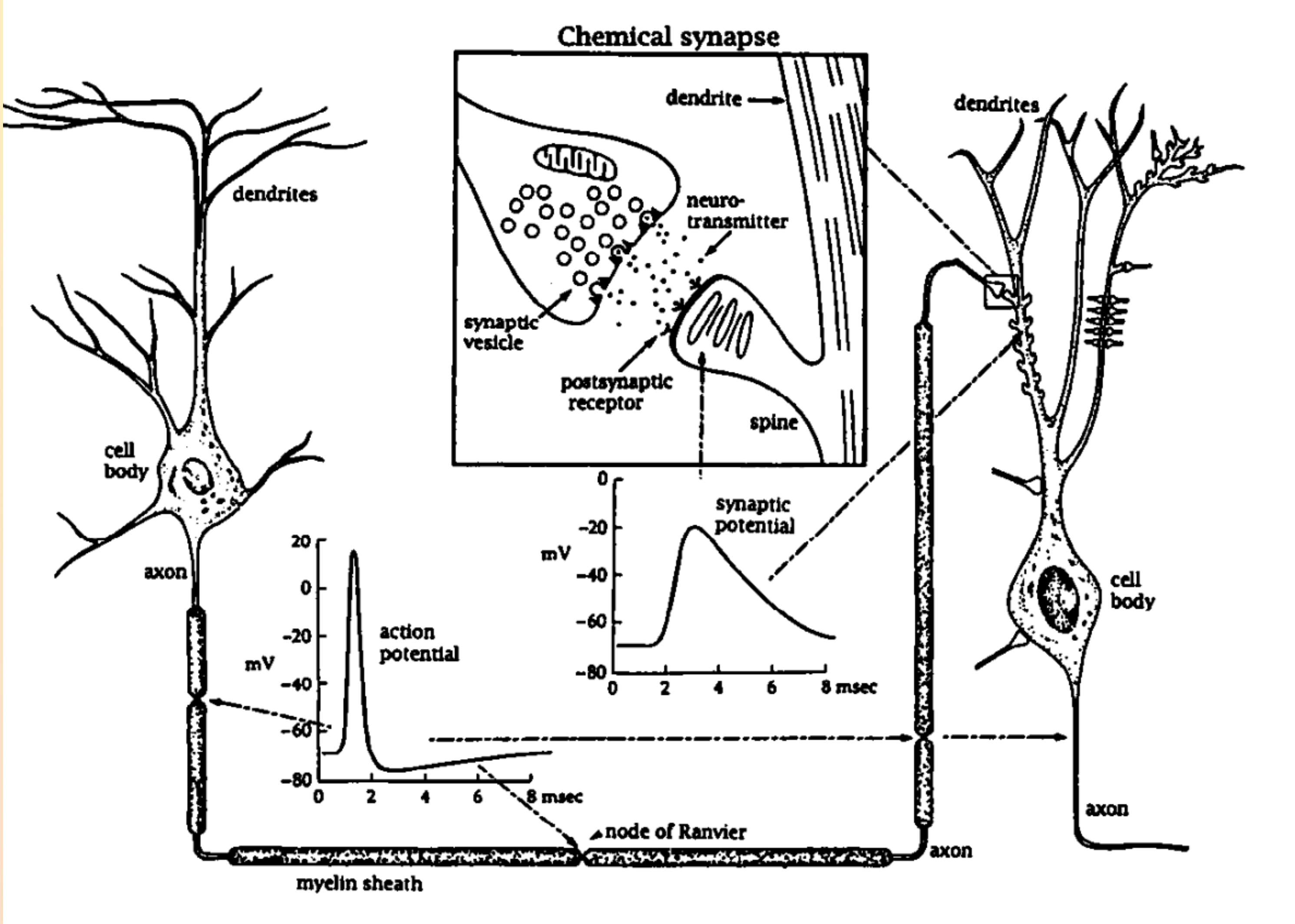
# Electroreceptor Organs

—
Image from Palave et al., 2022

# Electroreceptor Organs

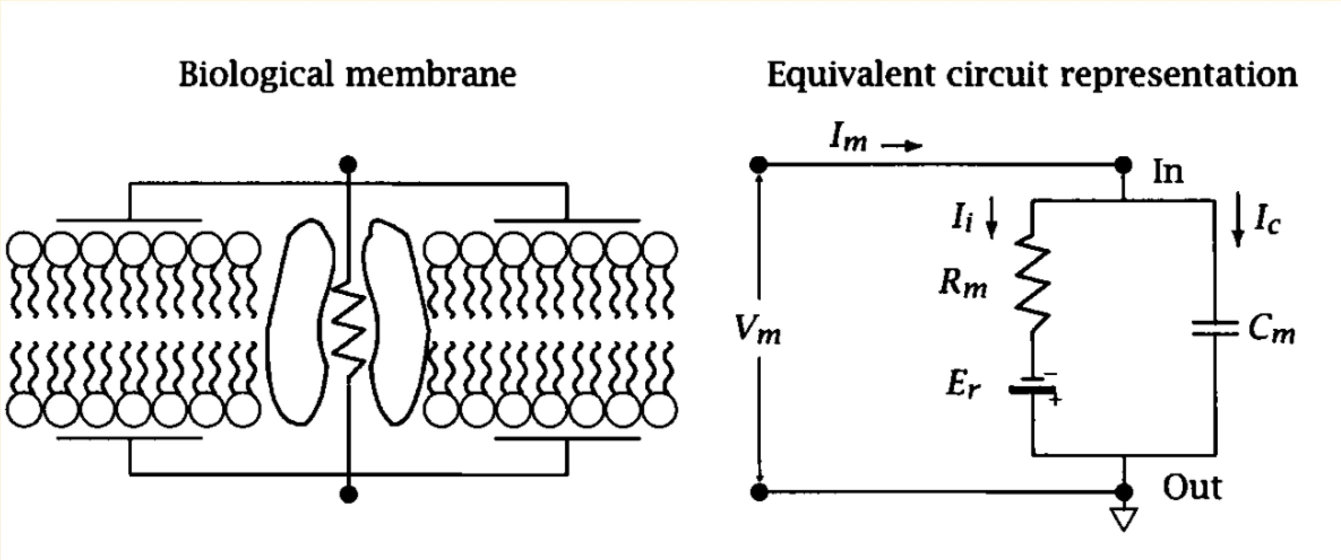


# Biological neurons



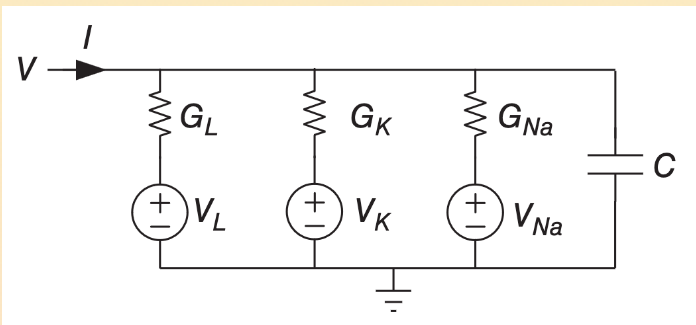

—
Image from Johnston, Wu (MIT Press, 1994)

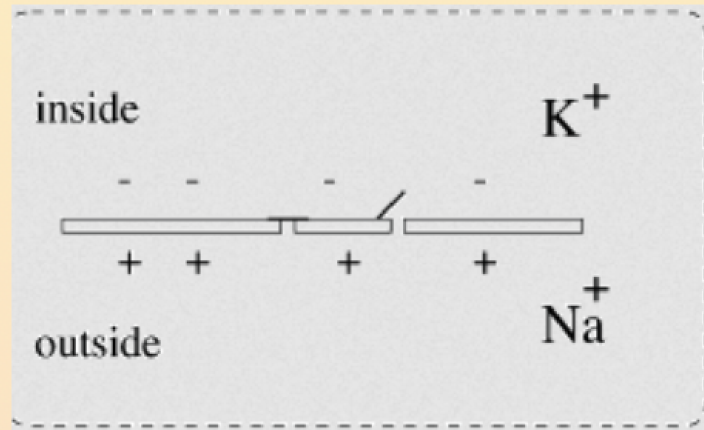
# Biological membranes as circuits



# The Hodgkin-Huxley Model

$$I = C\frac{dV}{dt} + G_{Na}m^3h(V - V_{Na}) + G_K n^4(V - V_K) + G_L(V - V_L)$$

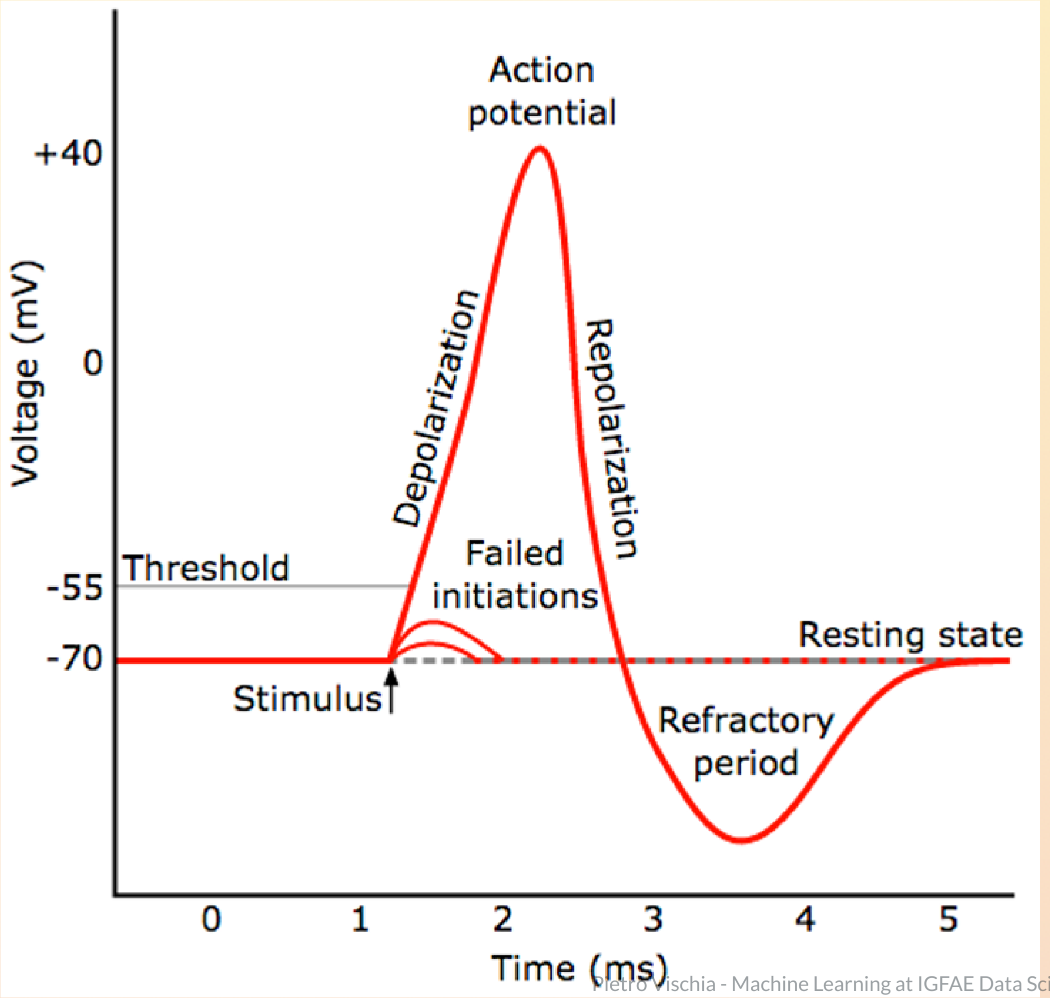



—
Image from Neuronal Dynamics and Zheng, Mazumder (Wiley, 2019)

# Spikes

- Information modulated in spatiotemporal patterns




—
Image from bioserver

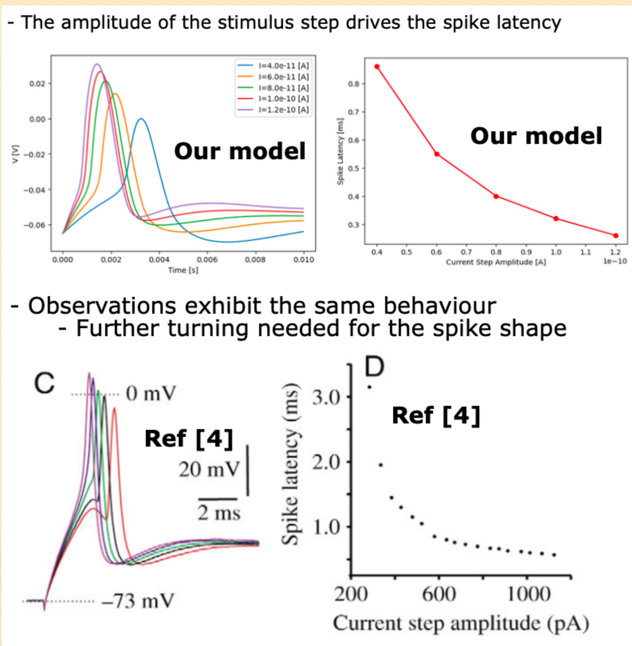
# Emergent phenomena?

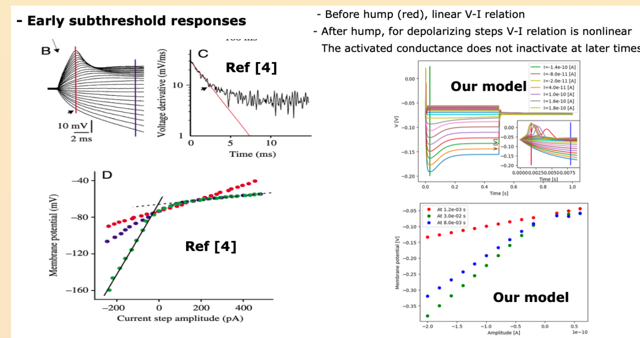
- Understand different levels of neuronal organization using computational models of neurons

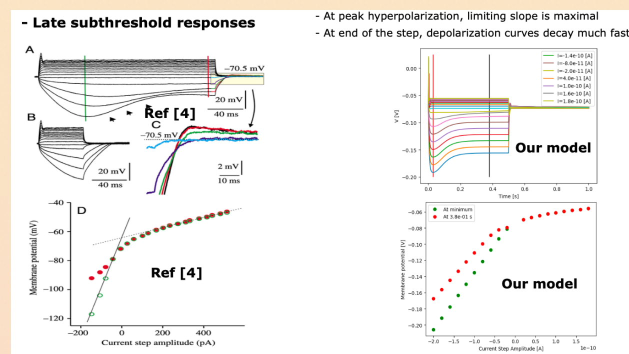

—
Image from utep.edu

# Neurons of Gymn. Om.

- Spherical neuron with four channels (different thresholds and time constants)
  - Vischia, Caputi 2023: computational model compared with data from "[4]" (J Exp Biol (2006) 209 (6): 1122–1134.)



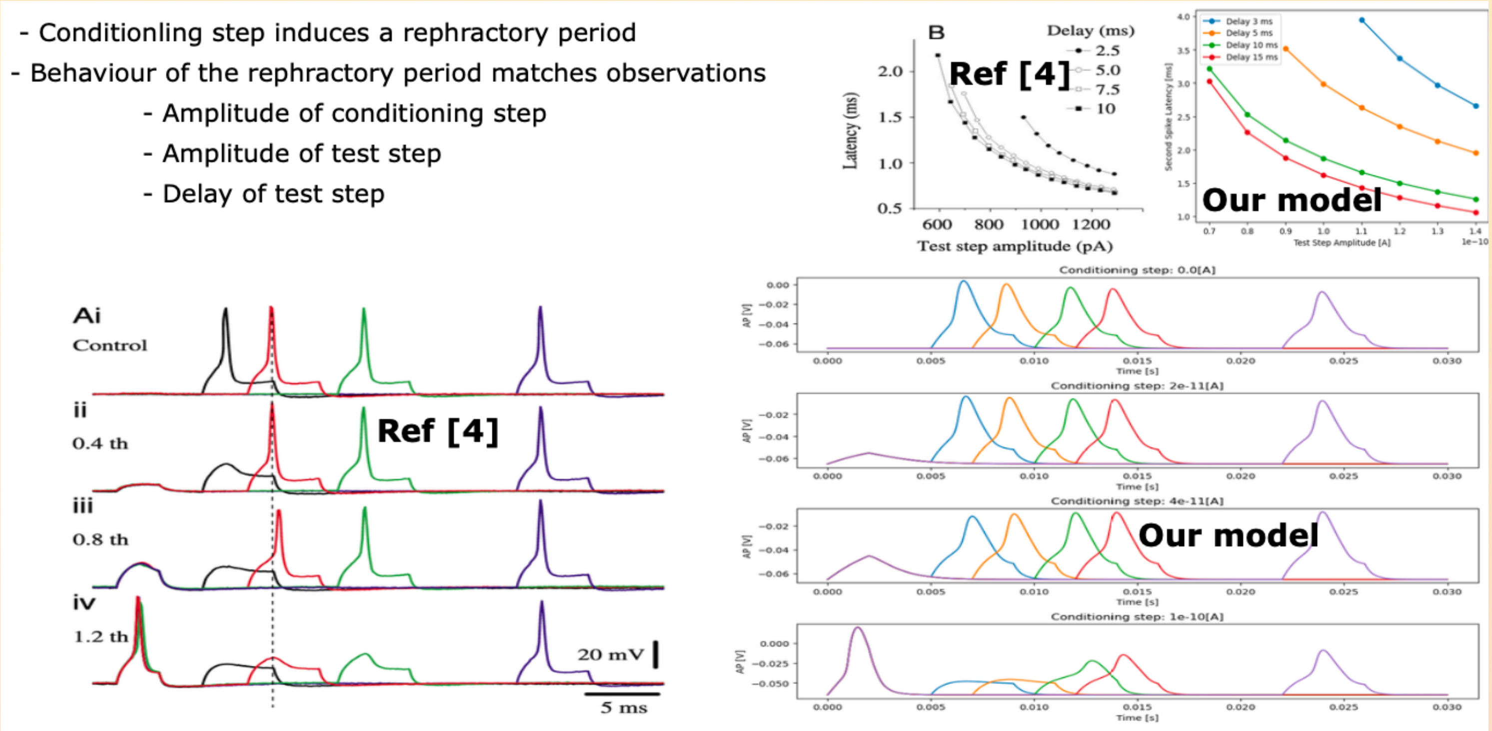




—

P. Vischia, A. Caputi, 10.5281/zenodo.8394819, paper in preparation

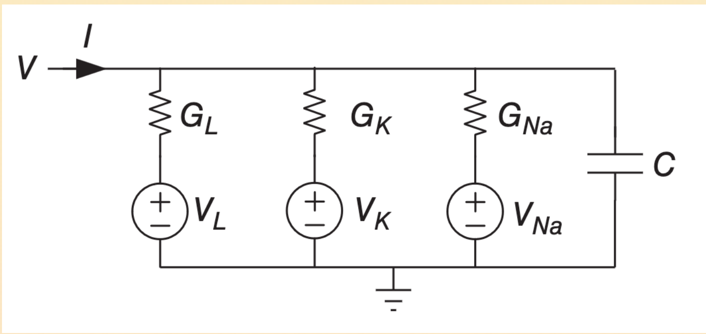
# Rephractory period

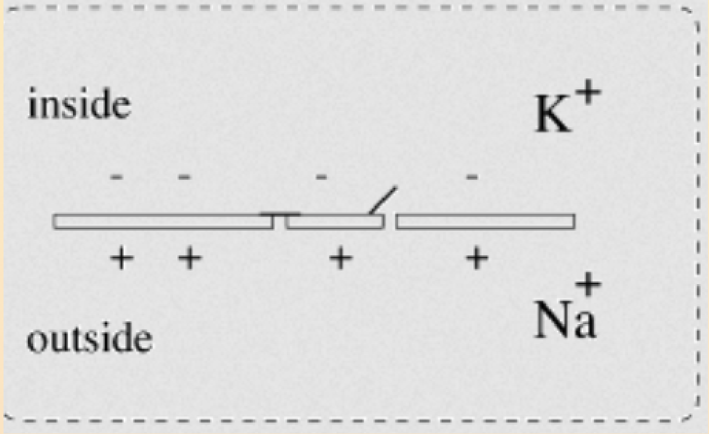
- Interaction between signals (potentially from different neurons)
- Next steps in preparation: study of emergent properties in the interaction between neurons



# Hodgkin-Huxley: good, but...

$$I = C\frac{dV}{dt} + G_{Na}m^3h(V - V_{Na}) + G_K n^4 (V - V_K) + G_L(V - V_L)$$



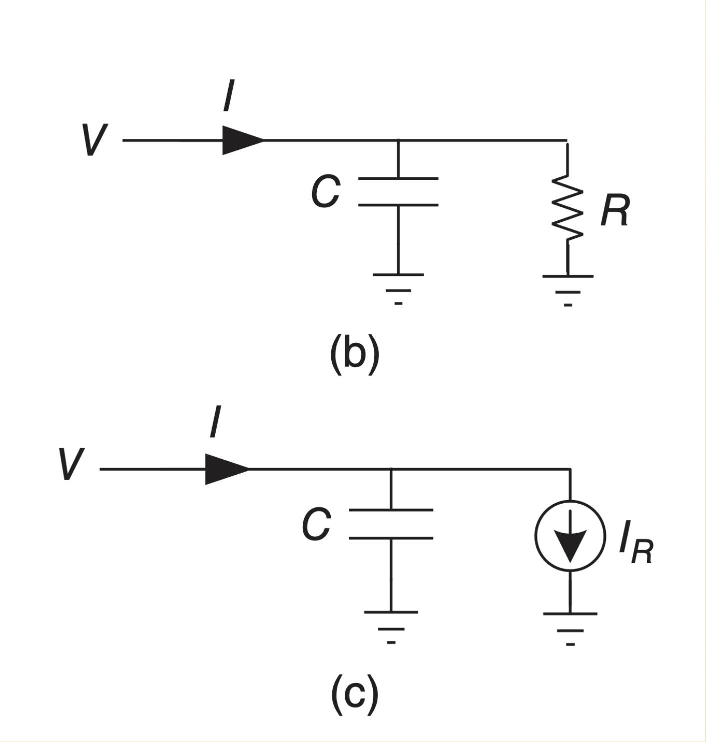




- Great to capture real neuron dynamics
- Computationally unfeasible for large networks

—
Image from Neuronal Dynamics and Zheng, Mazumder (Wiley, 2019)

# (Leaky) Integrate-and-fire Model

$$C\frac{dV(t)}{dt} = I - I_L$$



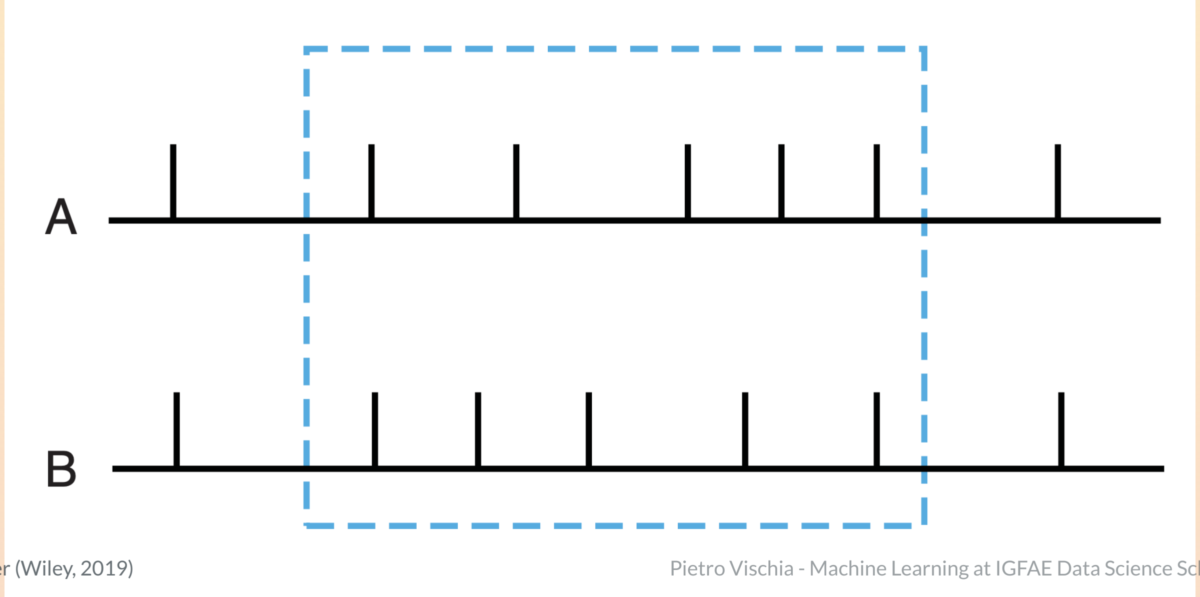

- Leakage current can be defined as:
    - Conductance-based: more plausible but high computational overhead, $I_L = f(V(t))$
    - Current-based: more computationally efficient

—

Image from Zheng, Mazumder (Wiley, 2019)

# Information encoding

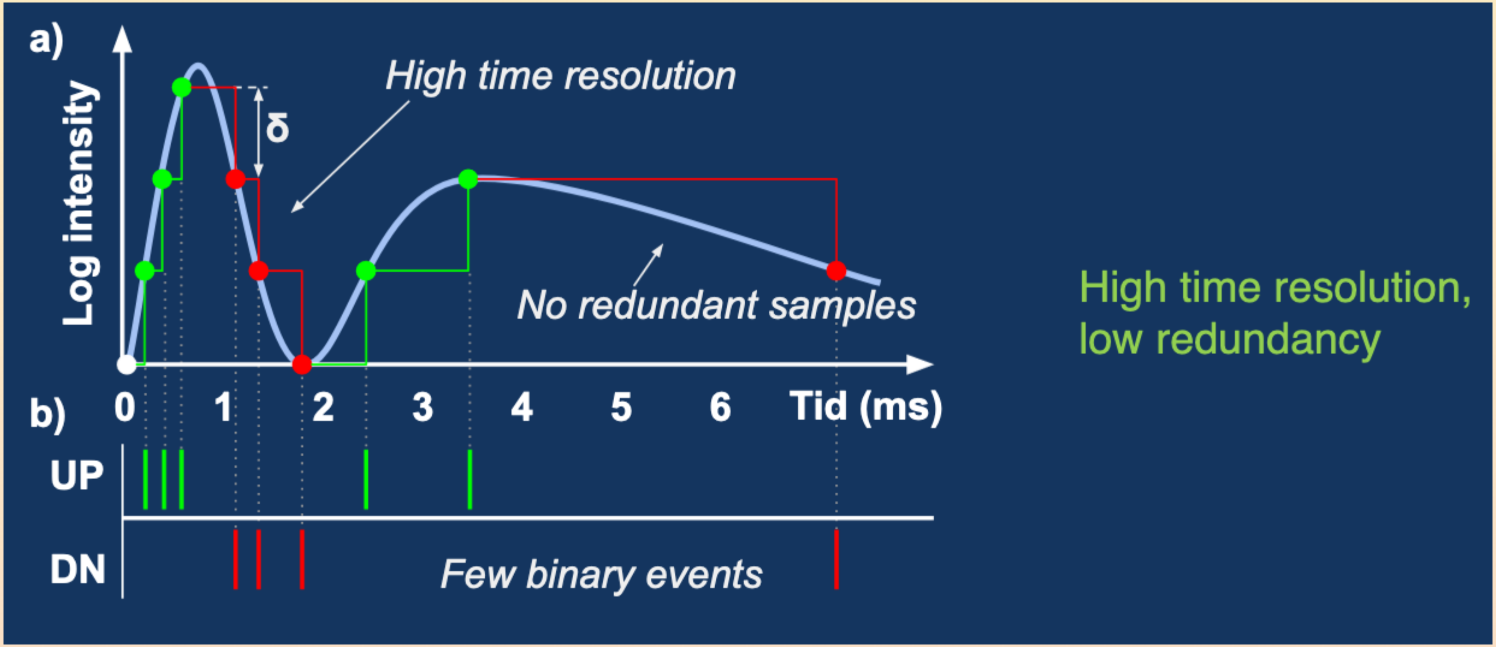
- Rate coding: information modulated on mean firing rate
  - Similar to frequency modulation in telecom
  - Low complexity $\longrightarrow$ easy encoding/decoding
  - Averaging $\longrightarrow$ Large response time (unrealistic), but noise-robustness
- Temporal coding: information carried by the exact timing of a spike
  - Feasibility supported by recent publications
  - Not noise-robust, therefore disfavoured by non-deterministic hardware implementations




—
Image from Zheng, Mazumder (Wiley, 2019)

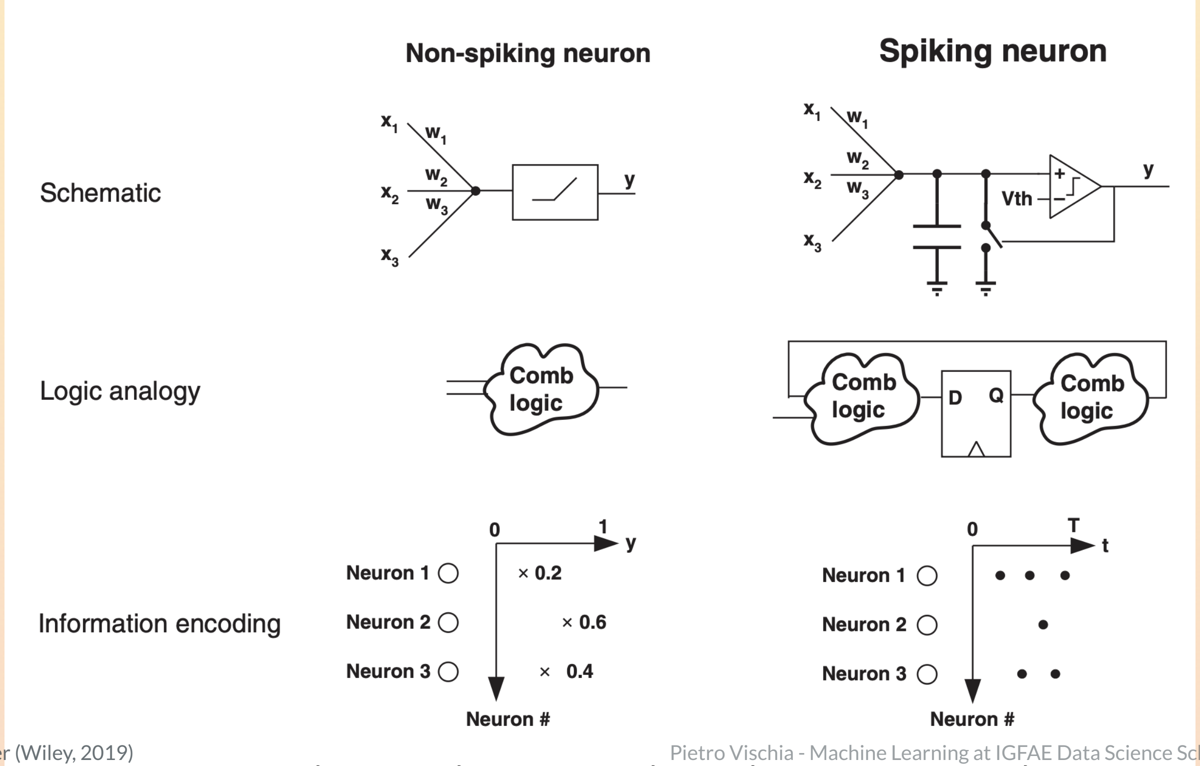
# Example of time encoding

- Efficient, threshold-based encoding
  - GPU (RTX3090): 40 GW simulation on discrete states
  - Human brain equivalent: 20 W on dynamic states



# Spike or not to spike

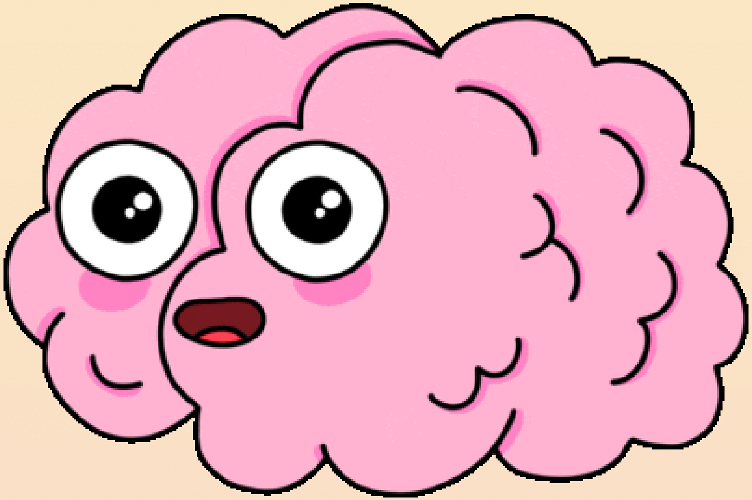
- Memory as intra-neuron capacitor
  - In perceptrons, memory can be implemented only via network structure (e.g. LSTM)
- Finite-state machine where output depends on previous history of inputs
- Encoding has a temporal distribution (good for spatio-temporal data)




—
Image from Zheng, Mazumder (Wiley, 2019)

# Learn with biological neurons

- Cannot use backpropagation-based gradient descent out of the box
  - Spikes are discrete in nature $\longrightarrow$ nondifferentiable
  - Temporal component makes things difficult

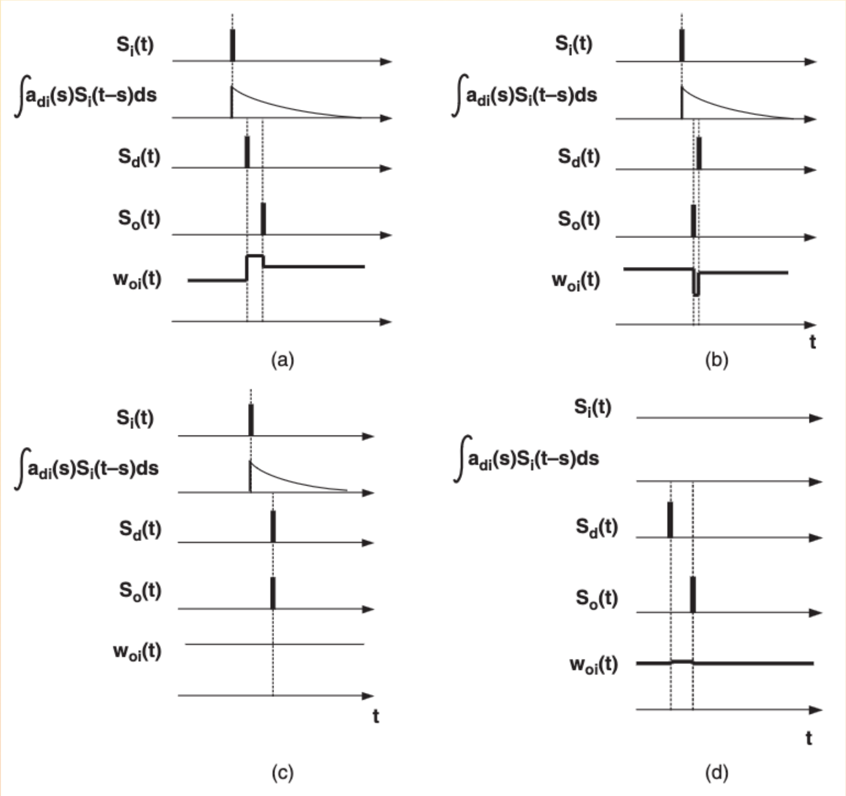


—
Image from giphy.com

# Example: ReSuMe

$$\frac{dw_{oi}}{dt} = (S_d(t) - S_o(t)\Big(a_d + \int_0^\infty a_{di}(s)S_i(t-s)ds\Big)$$

- Remote Supervised Method
  - Force network towards desired ($d$) spike trains
  - Potentiate $w$ when target spike
  - Depress $w$ when output spike
  - No change when $d$ and $o$ coincide
  - Weight change $\propto$ time difference
  - $a_d$ drive $o$ mean fire rate towards $d$ m.f.r.



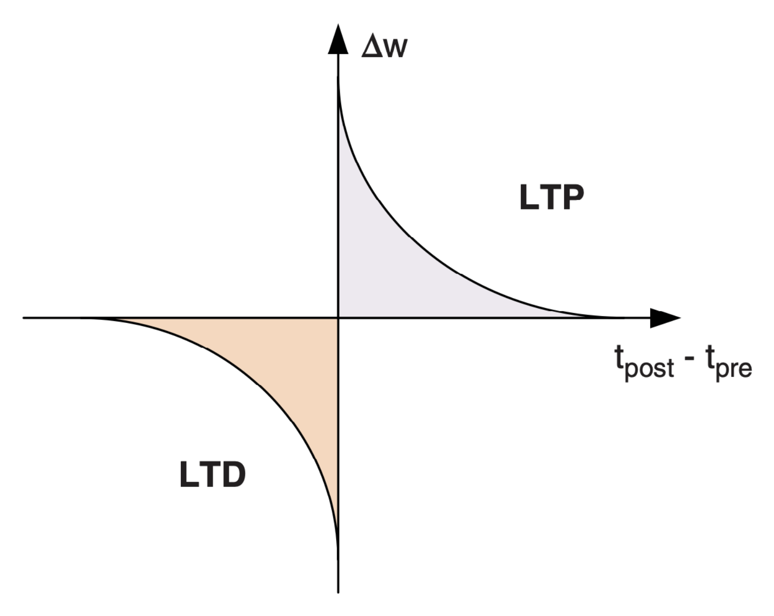

—
Image from Zheng, Mazumder (Wiley, 2019)

# Spike-Timing-Dependent Plasticity

- Hebbian rule: "in a sense, then, cells that fire together wire together"
    - But extended to account for relative timing of pre- and post-synaptic spikes

$$\Delta w = \sum_n \sum_m K(t^m_{post} - t^n_{pre})$$

- Excitation/Inhibition used to devise learning rules (supervised or unsupervised)

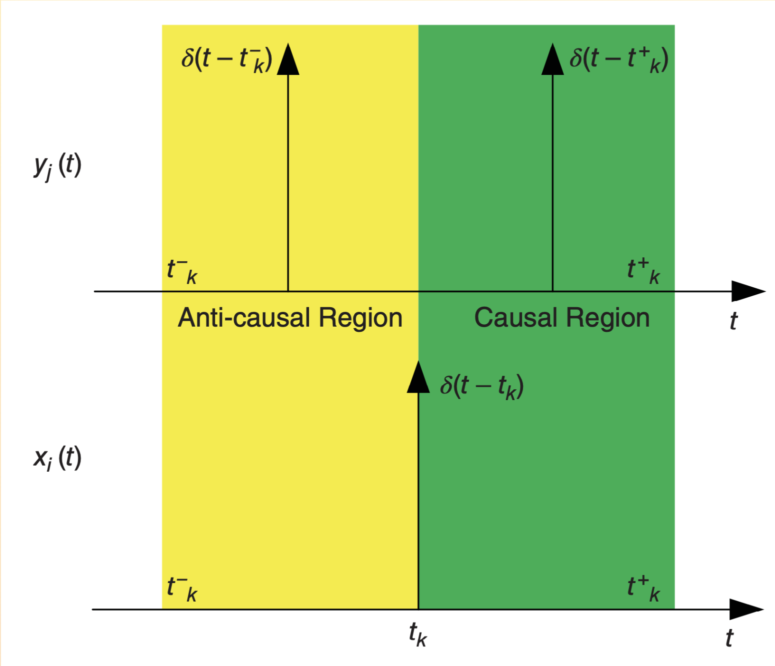



—
Image from Zheng, Mazumder (Wiley, 2019)

# Spike-Timing-Dependent Plasticity

$$\Delta w = \sum_n \sum_m K(t^m_{post} - t^n_{pre})$$

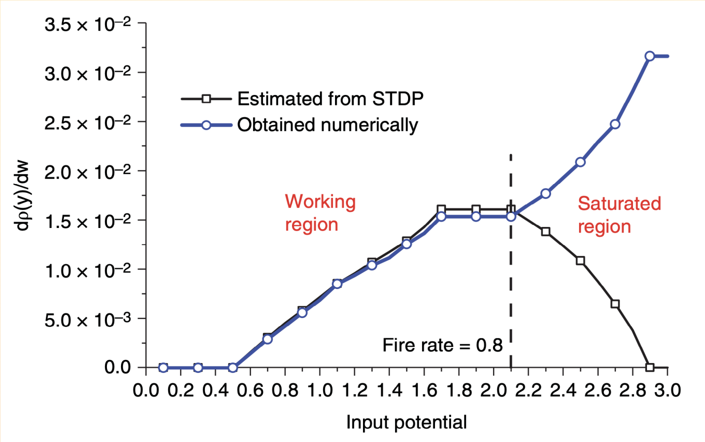
- Causal: Long-Term Potentiation.
  - Postsynaptic comes after presynaptic
- Anticausal: Long-Term Depression
  - Postsynaptic comes before presynaptic

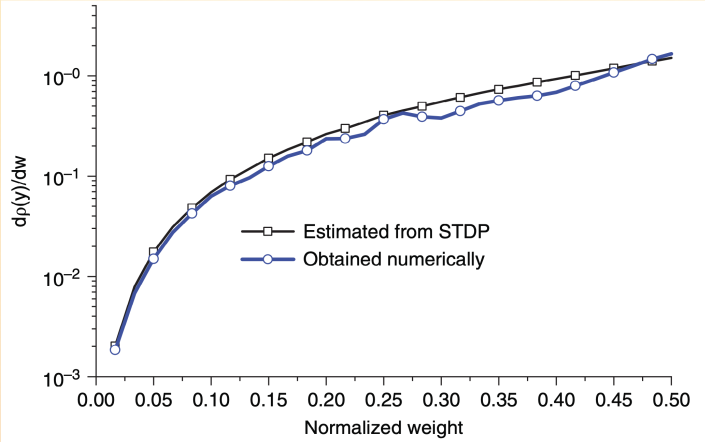



—
Image from Zheng, Mazumder (Wiley, 2019)

# Spike-Timing-Dependent Plasticity

- Gradients from STPD-based rules agree with numerical simulations
  - Except when firing rate is too high (it becomes difficult to estimate if input or output spike comes first)

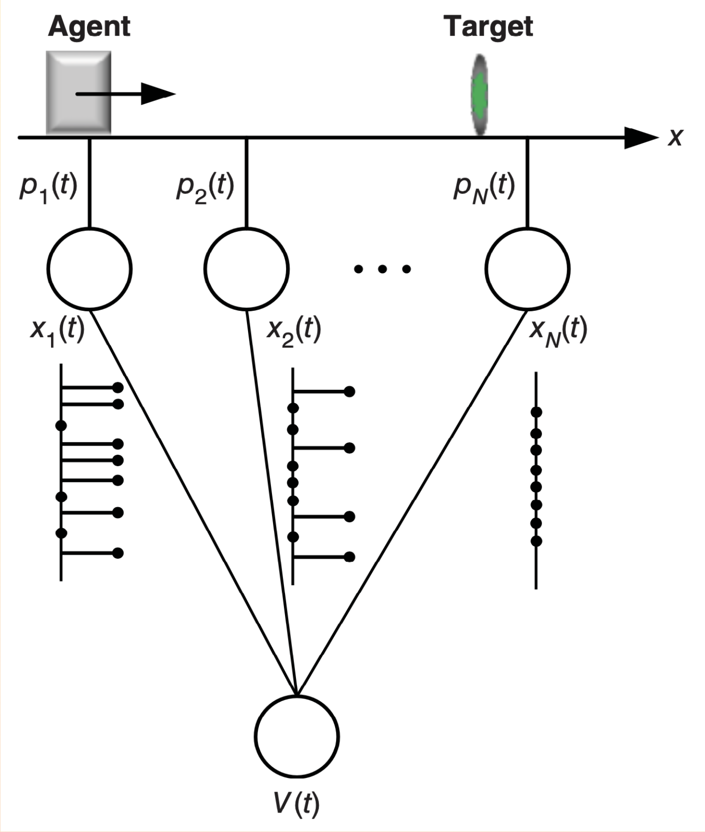


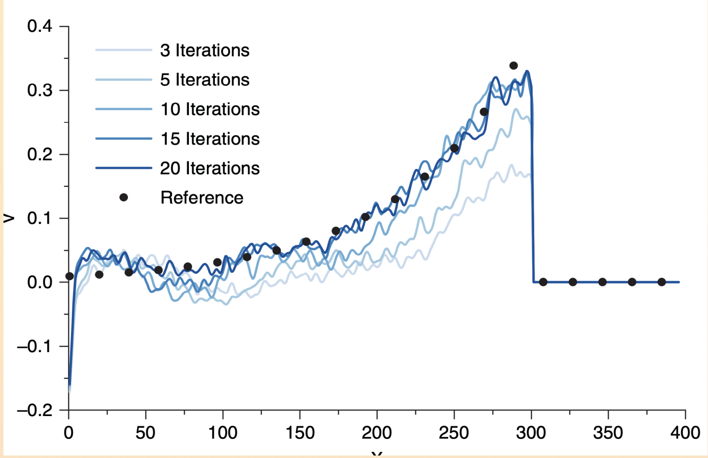




—
Image from Zheng, Mazumder (Wiley, 2019)

# STPD and reinforcement learning

- Neurons at different places fire when agent is close
- Agent is incentivised (rewarded) for being close to target



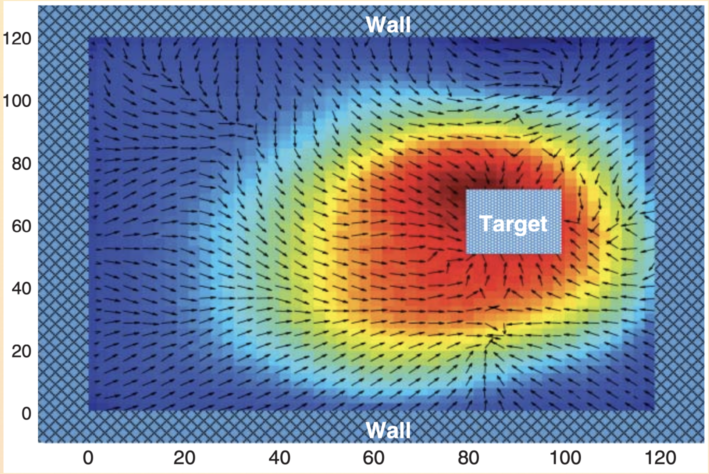



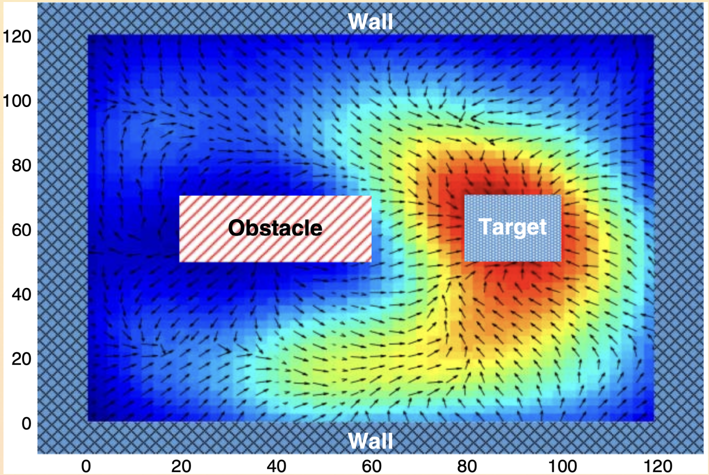

_

Image from Zheng, Mazumder (Wiley, 2019)

# STPD and reinforcement learning

- Example: maze
- Brighter colour $\longrightarrow$ agent thinks a reward is more likely



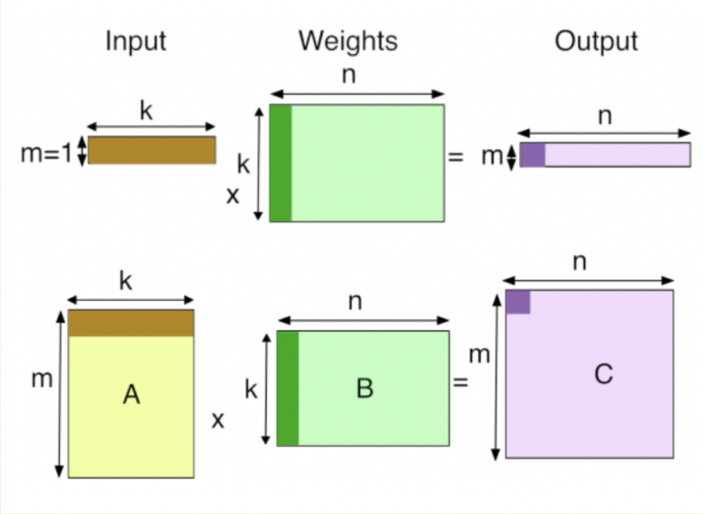

—
Image from Zheng, Mazumder (Wiley, 2019)

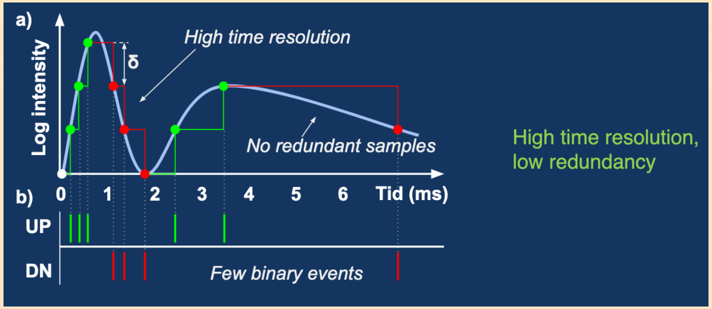
# Neuromorphic hardware

## The hardware implementation of spiking neural networks

# Different types of operations

- Perceptron-based networks: matrix multiplication
- Spiking neural networks: event-driven computations
    - "when a spike occurs, compute something"

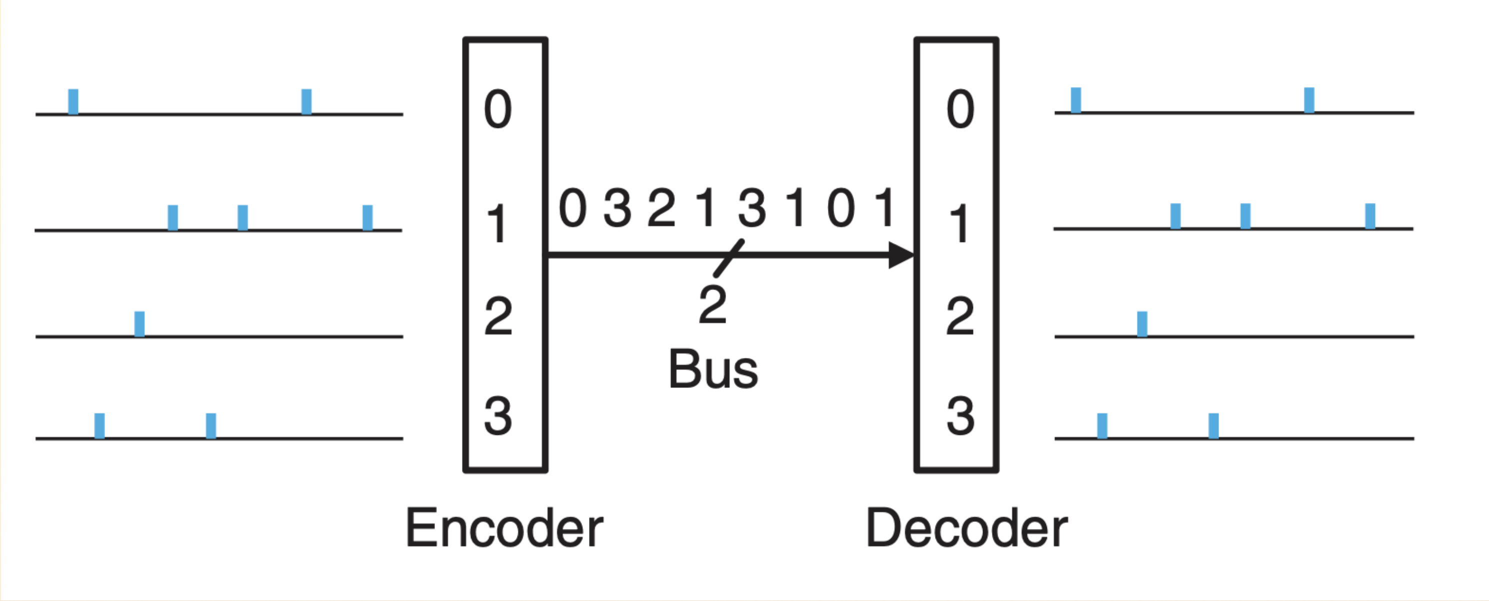



—
Figures by Gilles Louppe and Fredrik Sandin

# Address-Event Representation

- Gap in operational speed exploitable for time multiplexing
  - CMOS circuit operating speed $\mathcal{O}(ns)$
  - Neuromorphic system for real-time applications requires $\mathcal{O}(\mu s)$ or $\mathcal{O}(ms)$
- Vastly reduce routing complexity (number of physical interconnects)
  - Bus width: $N \rightarrow log_2 N$, where $N$ is the number of axons

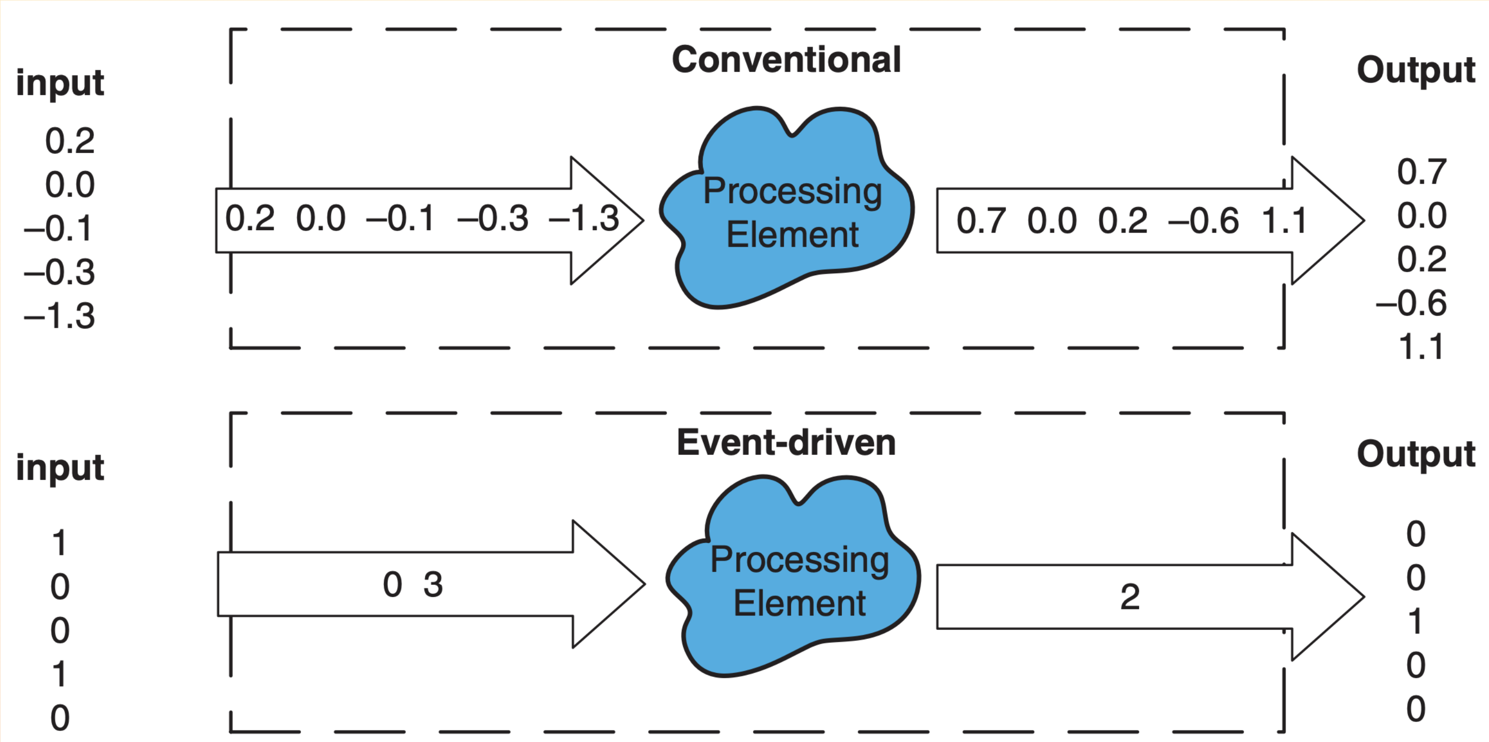



—

Image from Zheng, Mazumder (Wiley, 2019)

# The energy advantage

- Perceptron-based networks: matrix multiplication
  - Sparsity doesn't affect much the throughput and energy consumption
- Spiking neural networks: event-driven computations
  - Sparser inputs require less computations, therefore less time and energy

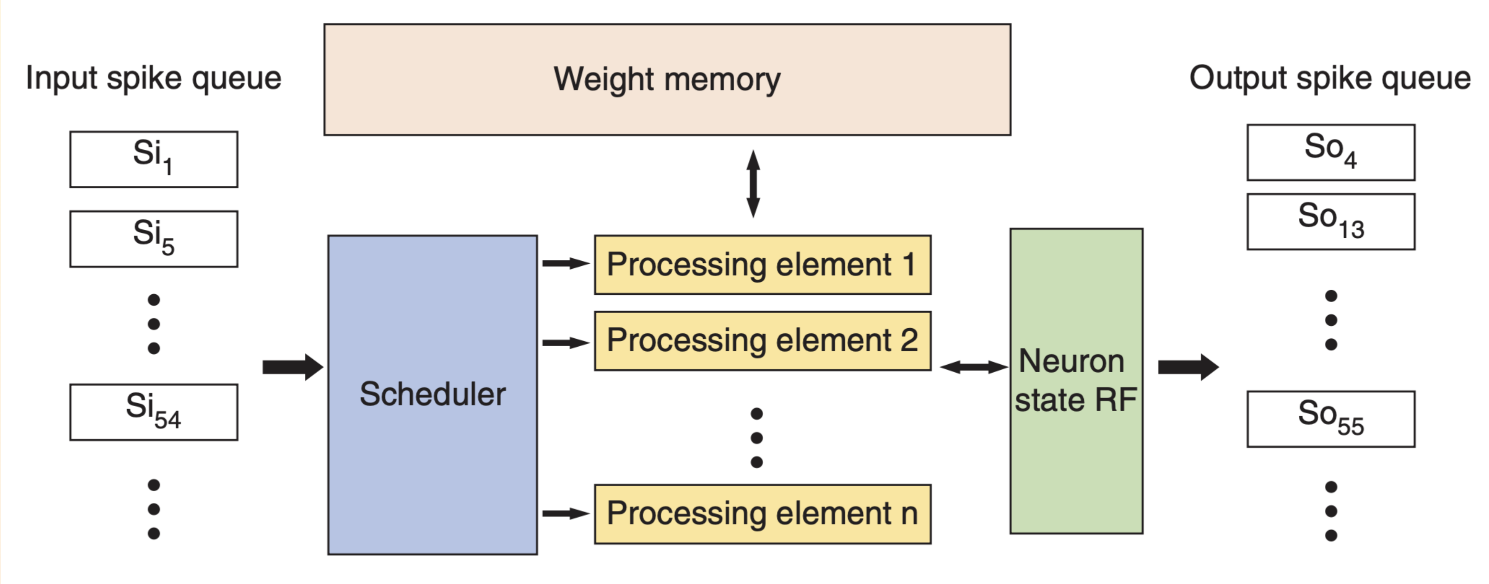



–

Image from Zheng, Mazumder (Wiley, 2019)

# Energy-efficient architectures

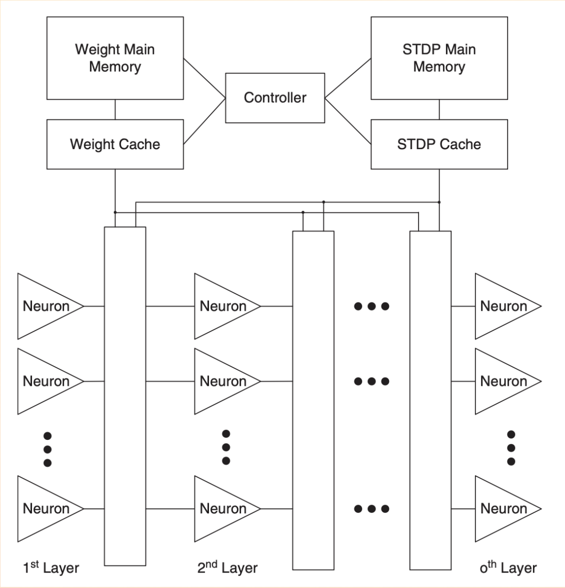
- Architectures allocate resources based on spike-generated computation requests

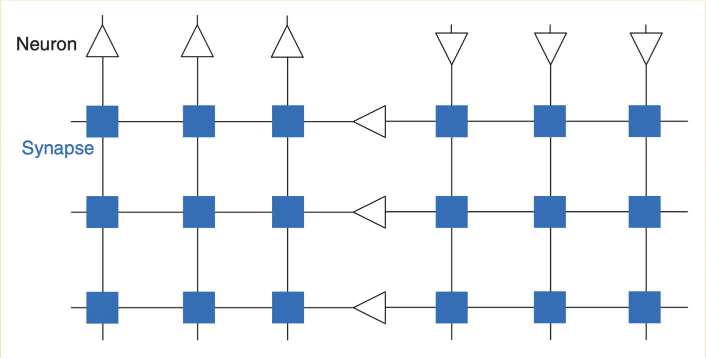



—
Image from Zheng, Mazumder (Wiley, 2019)

# STPD weight update in hardware

- Centralized architecture: memory and processing units are separate (similar to perceptron architectures)
  - Memory access cost is high
- Distributed memory architecture: can have in-memory processing $\longrightarrow$ high energy efficiency
  - Memory access cost is low




—

Image from Zheng, Mazumder (Wiley, 2019)

# Energy cost

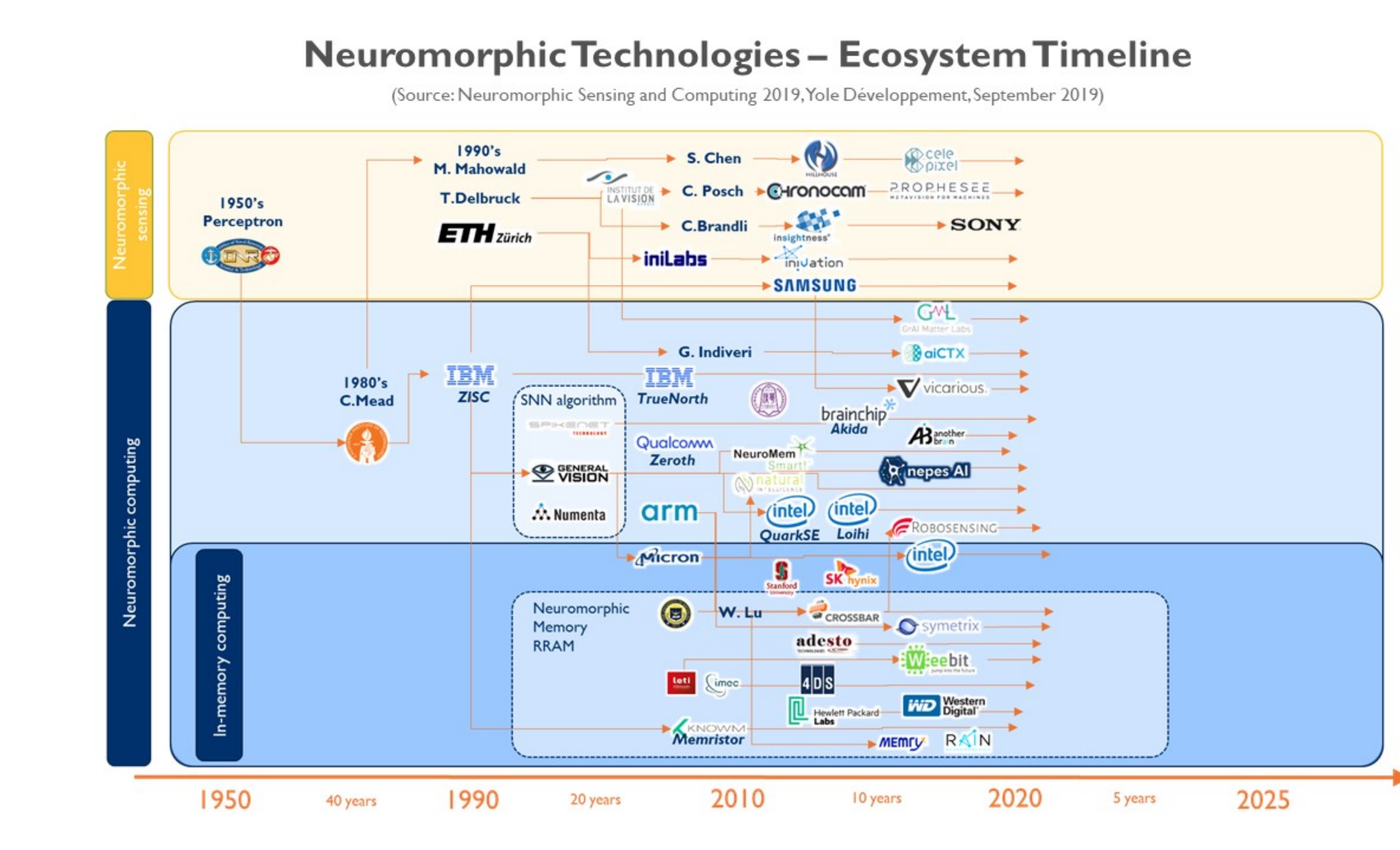
- Biological systems (human brains) still win, at the moment

| | Brain | Spikey | SpiNNaker | R2600X | Intel mobile | RTX2070 |
|---|---|---|---|---|---|---|
| Housekeeping | 4.75E-11 | 1.37E-06 | 1.66E-04 | 4.49E-04 | 1.23E-04 | **9.76E-07** |
| Resting potential | 5.77E-11 | **3.83E-08** | 8.99E-05 | 4.77E-05 | 4.25E-05 | 3.63E-06 |
| Action potential | 1.96E-11 | **4.39E-10** | 1.04E-08 | 3.04E-08 | 4.46E-09 | 4.71E-09 |
| Transmission | 8.17E-15 | **1.08E-11** | 9.59E-09 | 5.82E-08 | 2.14E-08 | 3.40E-09 |
| Single neuron | 2.49E-10 | **1.49E-06** | 3.33E-04 | 9.62E-04 | 3.37E-04 | 3.18E-05 |
| Full brain | 2.15E+01 | **1.29E+05** | 2.87E+07 | 8.29E+07 | 2.90E+07 | 2.74E+06 |

*Values for the simulation of 1s of model time are reported in Joule. The single neuron and full brain estimates assume a fan-out of 2,000 synapses and a spike rate of 4Hz. R2600X: AMD Ryzen 2600X. Intel mobile: Intel Core i7-4710MQ. RTX2070: NVIDIA RTX 2070. Both CPUs are measured using a PeakTech power meter. The lowest values from simulators/emulators are highlighted in bold.*

—
Table from 10.3389/fnins.2022.873935

# Implementations in history




—
Figure from Yolé Group

# Zoology of implementations

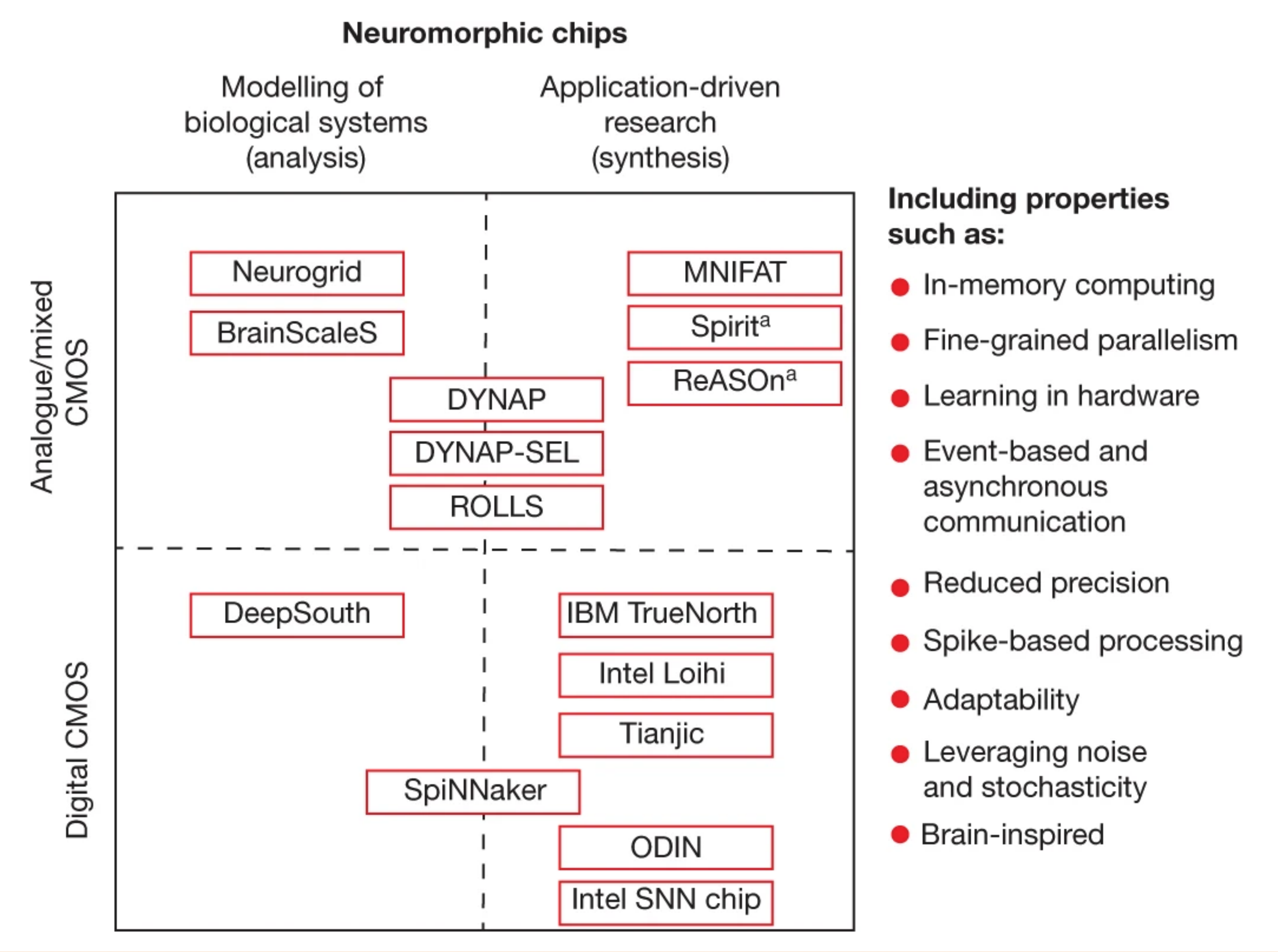



Figure from 10.1038/s41586-021-04362-w

# How to work on these systems

- Some of these chips come as kits
- Some have relatively easy companion software packages
- In particular, we will use Rockpool, an open-source tool designed by SynSense
    - Program and deploy on Dynap-SE2 and Xylo processors
    - Companion simulator (`xylosim`) provides estimates of expected energy consumption if algorithm is deployed on a real chip

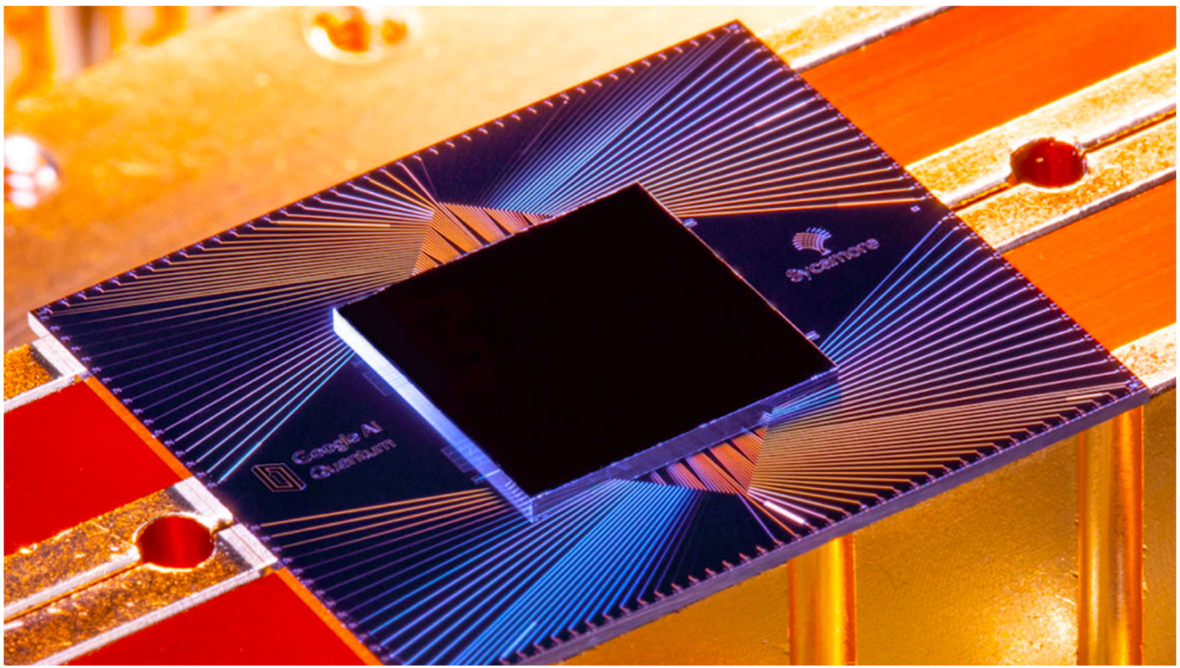
# Quantum Machine Learning

**Change the way information is encoded and treated**

# Quantum Supremacy?

## Google officially lays claim to quantum supremacy

A quantum computer reportedly beat the most powerful supercomputers at one type of calculation




Google researchers report that their quantum computer, Sycamore, has performed a calculation that can't be achieved with any classical computer. The quantum chip (shown) must be cooled to near absolute zero to function.

—

Article source and illustration from sciencenews.org

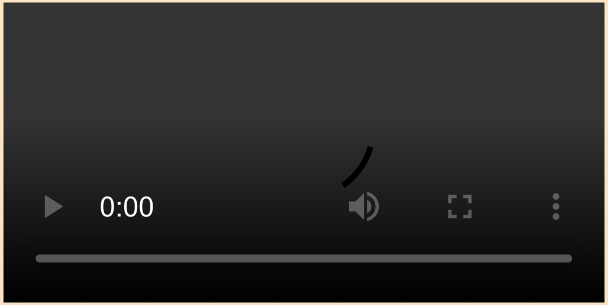
# Quantum and P-vs-NP

- Feynman goals when he introduced the concept of quantum computing:
  - To get an efficient way to simulate quantum mechanics
  - To make sure that quantum systems were at least capable of universal classical computation
- David Deutsch (QC pioneer, e.g. 10.1098/rspa.1985.0070)
  - To find an "empirical test" of the (controversial) Many-Worlds Interpretation of QM
  - To show that Nature has the property of computational universality (i.e., there's a single, programmable quantum system that can simulate any other quantum system).

—
From Scott Aaronson on quora

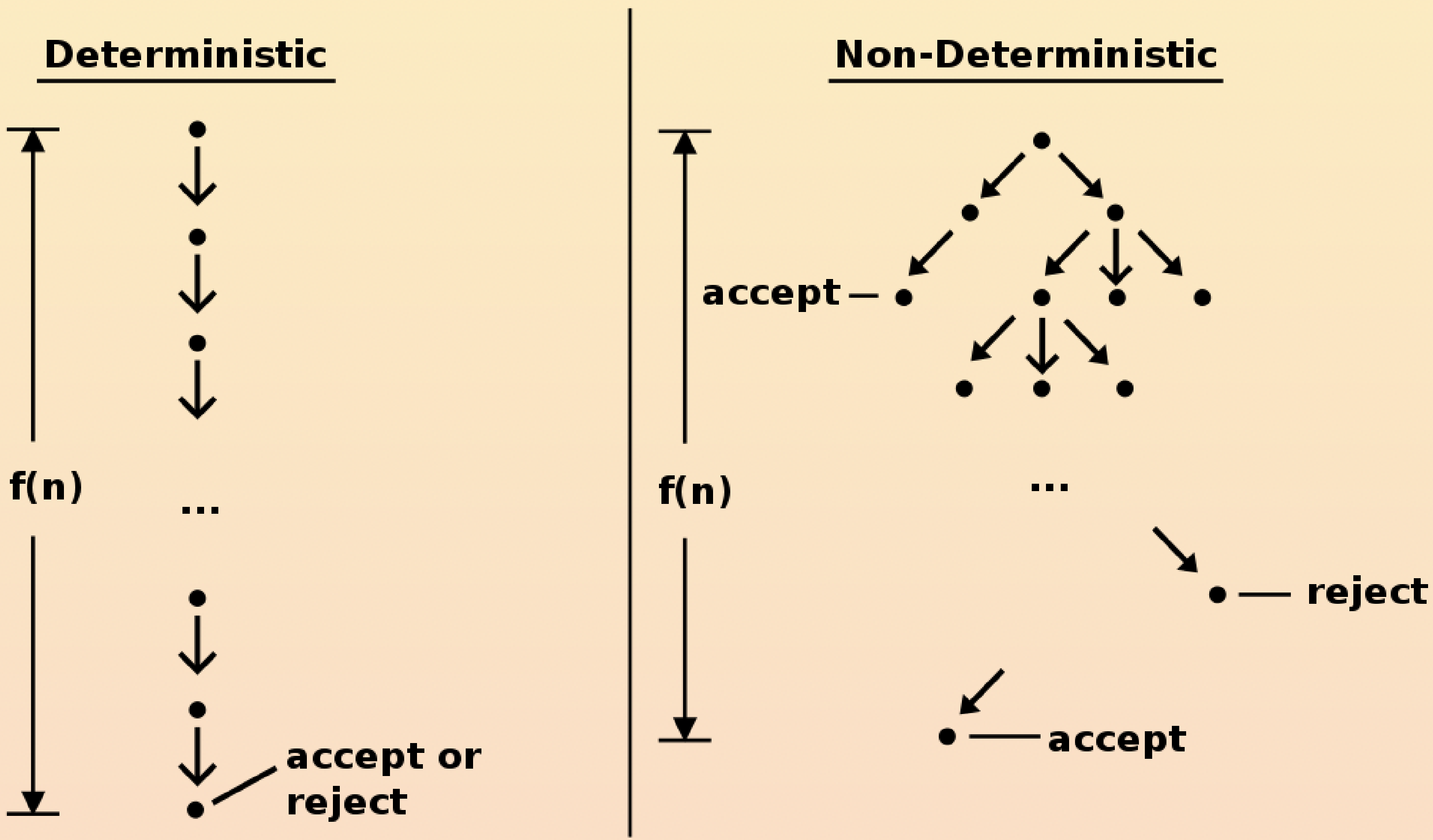
# Turing Machine

- Describe any computer algorithm using:
    - Infinite tape
    - Finite alphabet
    - Moving head
    - State register
    - Table of instructions




—
Video from youtube

# (Non-)Deterministic Turing Machine

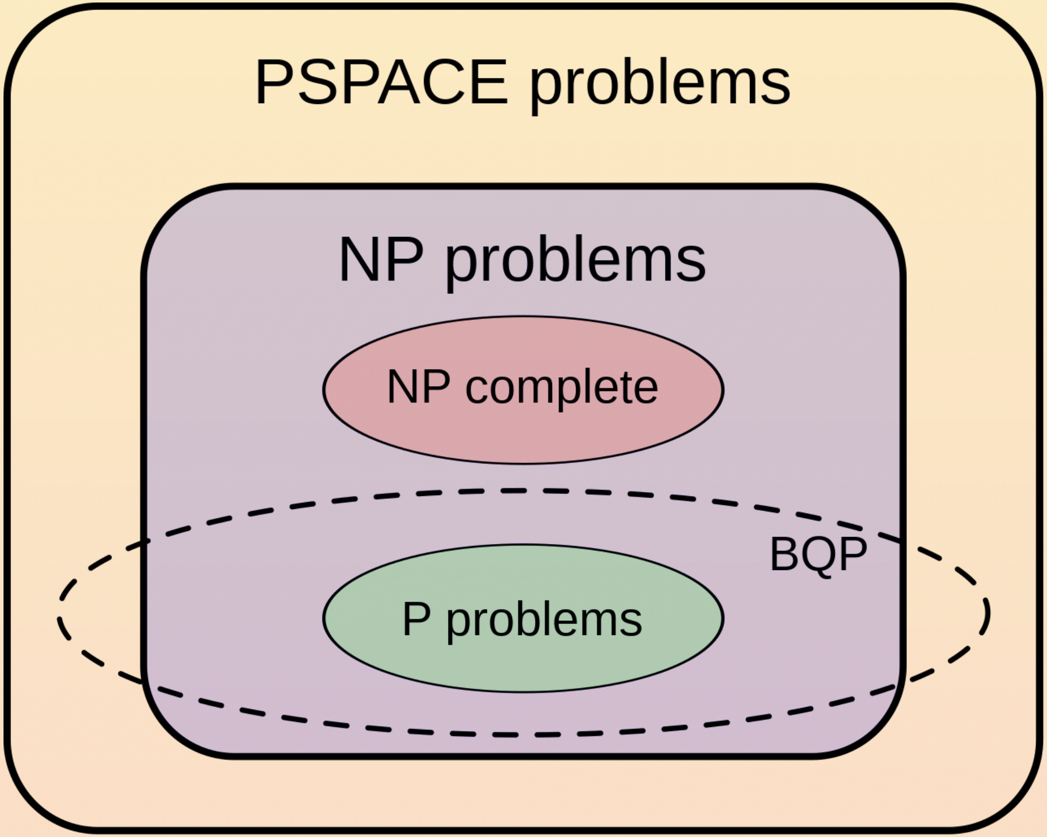



—

Image from wikipedia

# P vs NP

- Can problems that can be verified in polynomial time ($NP$) can also be solved in polynomial time ($P$)?




—
Image from wikipedia

# P=NP $\Rightarrow$ collapse of cryptography


## Crypto bro

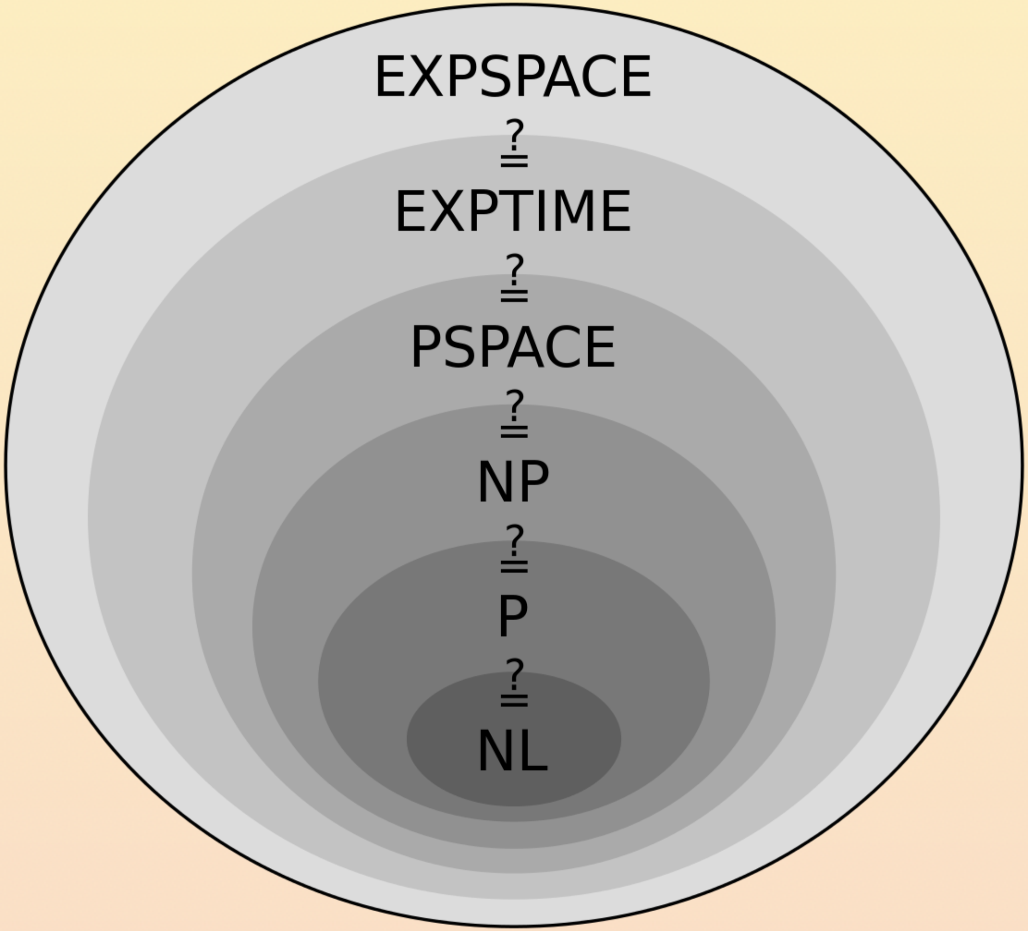
A person with a weak grasp on cryptocurrency/blockchain applications, yet has formed very strong opinions on the "best" ones. Often observed parading their involvement in crypto and arguing with other crypto bros.

*Crypto bro twitter bio: Entrepreneur. #HODL. $BTC. Living life in the clouds. Gym rat.*

**by VinixxiniV January 1, 2018**

Images from ascendex and Urban Dictionary

# Slightly more complex situation




—
Image from wikipedia

# Capacity

- Capacity: the upper bound to the number of bits that can be stored in the network during learning
    - Transfer of (Fisher or Shannon) information from the training data to the weights of the synapses
    - Related to the number of trainable parameters

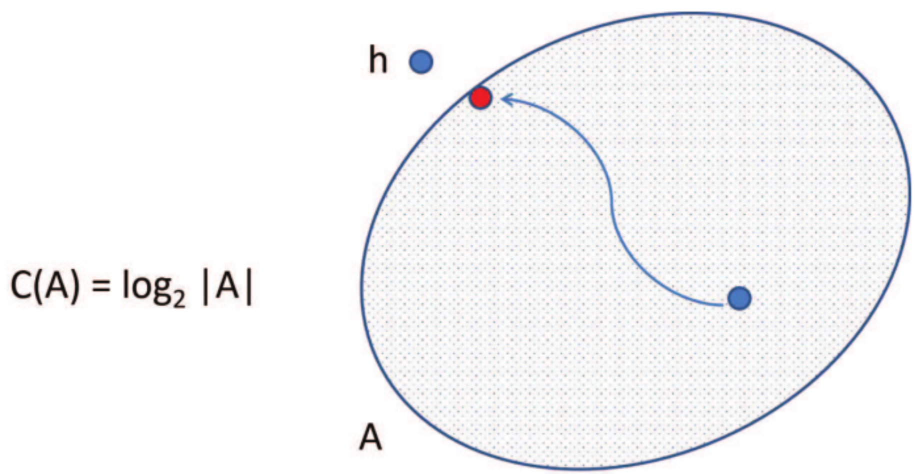
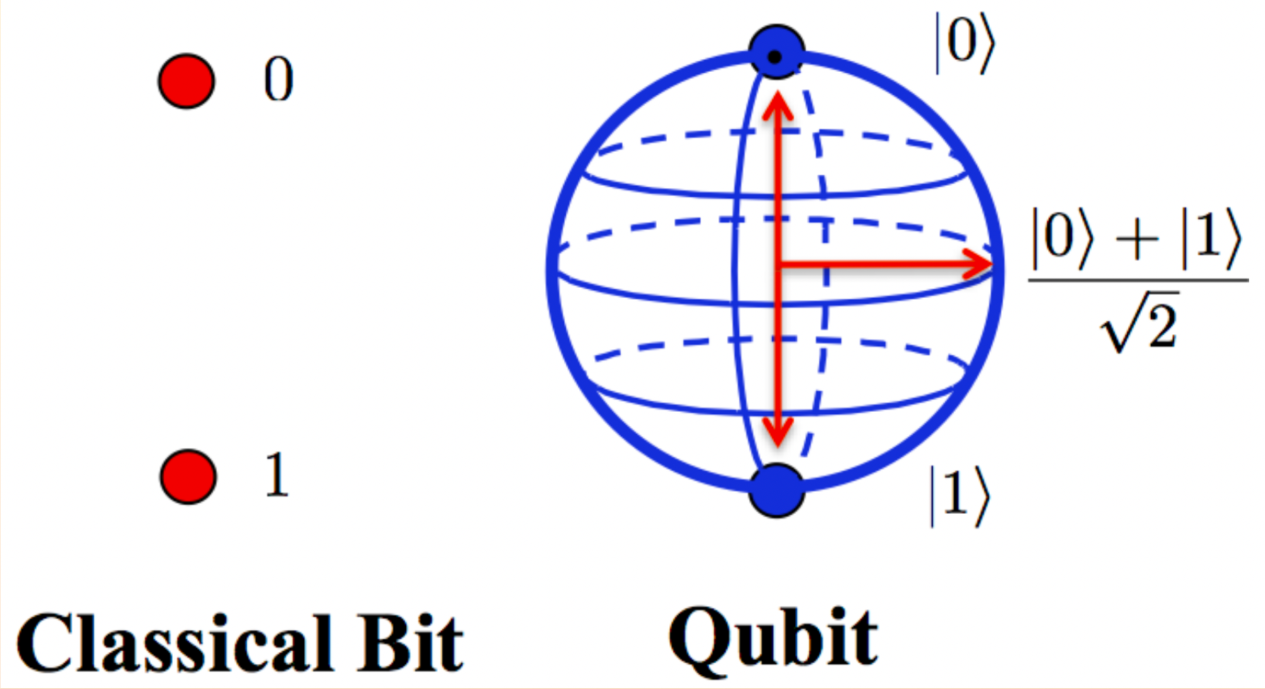



Figure 1. Learning framework where $h$ is the function to be learnt and $A$ is the available class of hypothesis or approximating functions. The cardinal capacity is the logarithm base two of the number, or volume, of the functions contained in $A$.

—
Figure from Baldi and Vershynin, 1901.00434

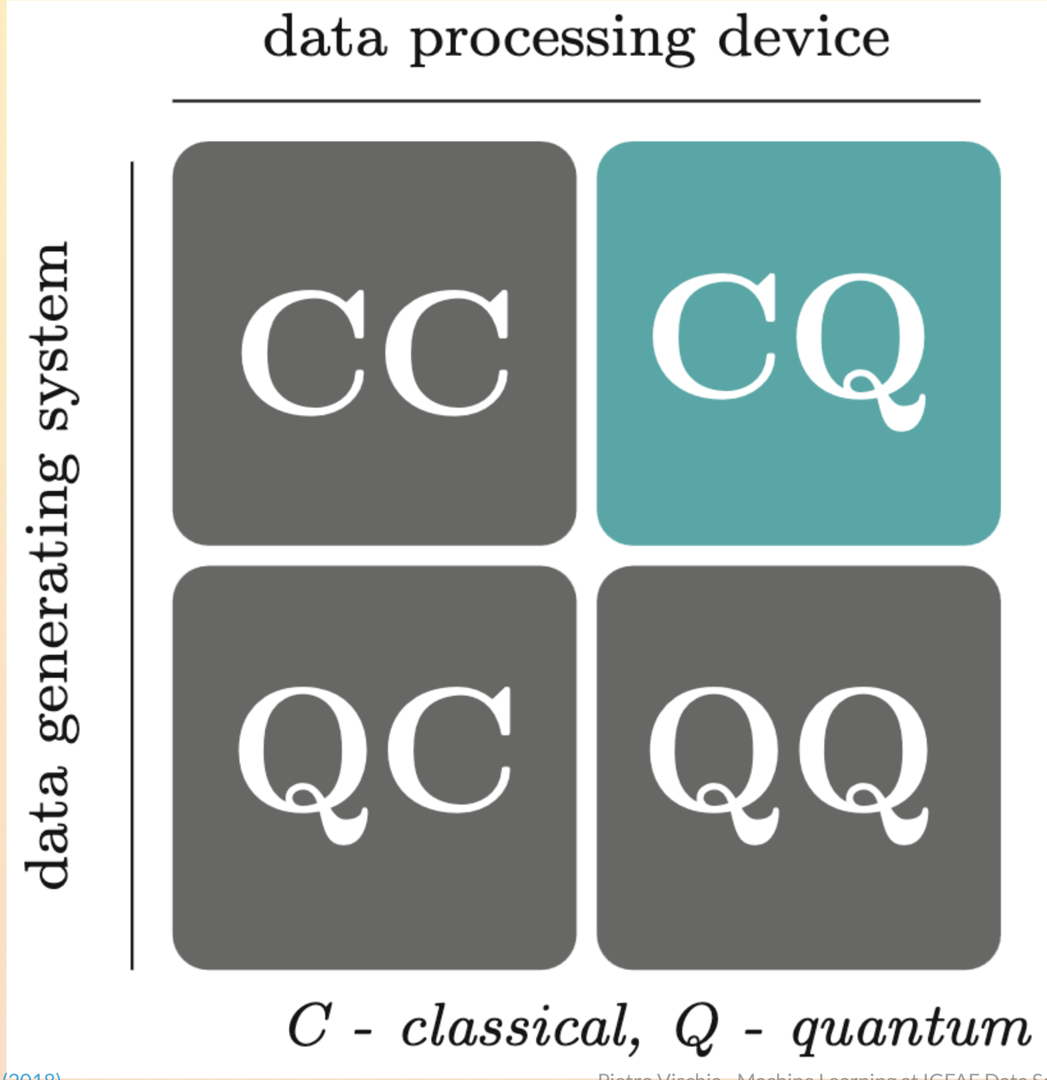
# Encode information with Qubits

- Random bit (Bernoulli random variable) whose description is not governed by classical probability theory but by quantum mechanics
- Not only "because it can take real values in $[0, 1]$": complex numbers as coefficients $\alpha$ and $\beta$ create interference
    - Interference is not reproducible with classical bits




—
Image by prateekvjoshi

# Quantum Machine Learning




—

Image from M. Schuld, F. Petruccione (2018)

# Two coins

## Classical coin

| State | Preparation | Toss coin 1 | Toss coin 1 again |
|---|---|---|---|
| (heads, heads) | 1 | 0.5 | 0.5 |
| (heads, tails) | 0 | 0 | 0 |
| (tails, heads) | 0 | 0.5 | 0.5 |
| (tails, tails) | 0 | 0 | 0 |

## Quantum coin

| State | Preparation | Toss coin 1, don't observe | Toss coin 1 again |
|---|---|---|---|
| $\|heads > \|heads >$ | 1 | 0.5 | 1 |
| $\|heads > \|tails >$ | 0 | 0 | 0 |
| $\|tails > \|heads >$ | 0 | 0.5 | 0 |
| $\|tails > \|tails >$ | 0 | 0 | 0 |

# Classical probability

Probability vectors:

$$p = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \longrightarrow p' = \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}$$

Transition probability matrix (lines sum up to one, 3rd axiom of Kolmogorov):

$$S = \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$p' = Sp = \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}, \qquad p'' = Sp' = \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}$$

# Qbit

Amplitude vectors $\alpha = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, with probability $p = |\alpha|^2$

$$\alpha = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \longrightarrow p' = \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}$$

Transition via unitary ($U^\dagger U = UU^\dagger = I$) complex matrix (Hadamard gate):

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

Negative entries induce laws different than the ones in classical probability theory!

# Qbit and interference

$$\alpha' = H\alpha = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\1\\0\end{bmatrix}$$, with probabilities:

$$p(|heads > |heads >) = p(|tails > |heads >) = |\sqrt{0.5}|^2 = 0.5$$

Now applying the transformation again:

$$\alpha'' = H\alpha' = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\0 & 1 & 0 & 1\\1 & 0 & -1 & 0\\0 & 1 & 0 & -1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\1\\0\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1+0+1+0\\0+0+0+0\\1+0-1+0\\0+0+0+0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$$

From a state of high uncertainty to a state of lower uncertainty: counterintuitive!

# What if I observe after the first measurement?

$$\alpha'_{obs} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$, or $$\alpha'_{obs} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

and in both cases:

$$\alpha'' = H\alpha'_{obs} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$, with probabilities:

$$p(|heads > |heads >) = p(|tails > |heads >) = |\sqrt{0.5}|^2 = 0.5$$

# When observing intermediate state

Recover classical picture

| State | Preparation | Toss coin 1 and observe | Toss coin 1 again |
|---|---|---|---|
| $\|heads > \|heads >$ | 1 | 0.5 | 0.5 |
| $\|heads > \|tails >$ | 0 | 0 | 0 |
| $\|tails > \|heads >$ | 0 | 0.5 | 0.5 |
| $\|tails > \|tails >$ | 0 | 0 | 0 |

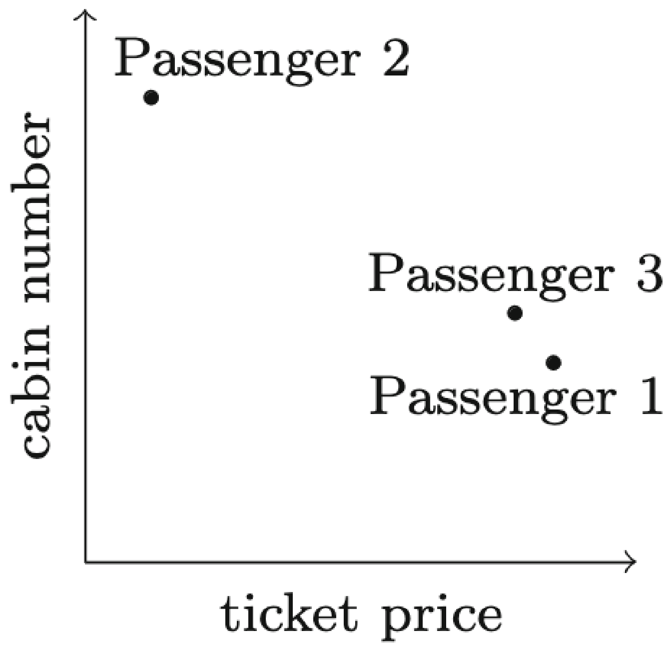
# Quantum computer and algorithms

- Quantum computer: physical implementation of $N$ qubits, with precise control on the evolution of the state
- Quantum algorithm: targeted manipulation of the quantum system, with a subsequent measurement to retrieve information from the system
- Quantum computers are sampling devices
  - Choose experimental configuration (e.g. strenght of a magnetic field)
  - Read out a distribution over all possible measurement outcomes

# Prepare the data

## Classical

**Fig. 1.2** The mini-dataset displayed in a graph. The similarity (Euclidean distance) between Passengers 1 and 3 is closer than between Passengers 2 and 3




—

Image from M. Schuld, F. Petruccione (2018)

# Prepare the data

## Quantum

| | price | room | survival |
|---|---|---|---|
| Passenger 1 | 0.921 | 0.390 | yes (1) |
| Passenger 2 | 0.141 | 0.990 | no (0) |
| Passenger 3 | 0.866 | 0.500 | ? |

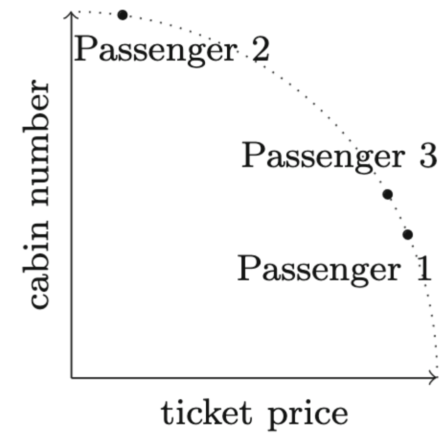



**Fig. 1.3** Left: Additional preprocessing of the data. Each feature vector gets normalised to unit length. Right: Preprocessed data displayed in a graph. The points now lie on a unit circle. The Euclidean distance between Passengers 1 and 3 is still smaller than between Passengers 2 and 3

—

Image from M. Schuld, F. Petruccione (2018)

# Data encoding

- Represent data in terms of Qubits

**Table 3.3** Some useful single qubit logic gates and their representations

| Gate | Circuit representation | Matrix representation | Dirac representation |
|---|---|---|---|
| $X$ | $-\boxed{X}-$ | $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ | $\|1\rangle\langle 0\| + \|0\rangle\langle 1\|$ |
| $Y$ | $-\boxed{Y}-$ | $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ | $i\|1\rangle\langle 0\| - i\|0\rangle\langle 1\|$ |
| $Z$ | $-\boxed{Z}-$ | $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ | $\|1\rangle\langle 0\| - \|0\rangle\langle 1\|$ |
| $H$ | $-\boxed{H}-$ | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ | $\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)\langle 0\| +$ $\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)\langle 1\|$ |
| $S$ | $-\boxed{S}-$ | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ | $\frac{1}{\sqrt{2}}\|0\rangle\langle 0\| + \frac{1}{\sqrt{2}}i\|1\rangle\langle 1\|$ |
| $R$ | $-\boxed{R}-$ | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & \exp(-i\pi/4) \end{pmatrix}$ | $\frac{1}{\sqrt{2}}\|0\rangle\langle 0\| +$ $\frac{1}{\sqrt{2}}\exp^{-i\pi/4}\|1\rangle\langle 1\|$ |

—
Image from M. Schuld, F. Petruccione (2018)

# Evolution of a quantum system

$$|\psi(t)>= U(t)|\psi(0)>$$

- $U(t)$ is a unitary operator (matrix)
- Qubit $|\psi>= \alpha|0> + \beta|1>$
- Gates: action, representable by unitary matrix, on one or two qubits

# Evolution of a quantum system

- Approximate with a sequence of a small dictionary of quantum gates acting on one or two qubits at the time
- Circuit of gates: same as for classical algos, but built upon a limited number of logic gates
- All operations are reversible (consequence of unitarity)

**Table 3.3** Some useful single qubit logic gates and their representations

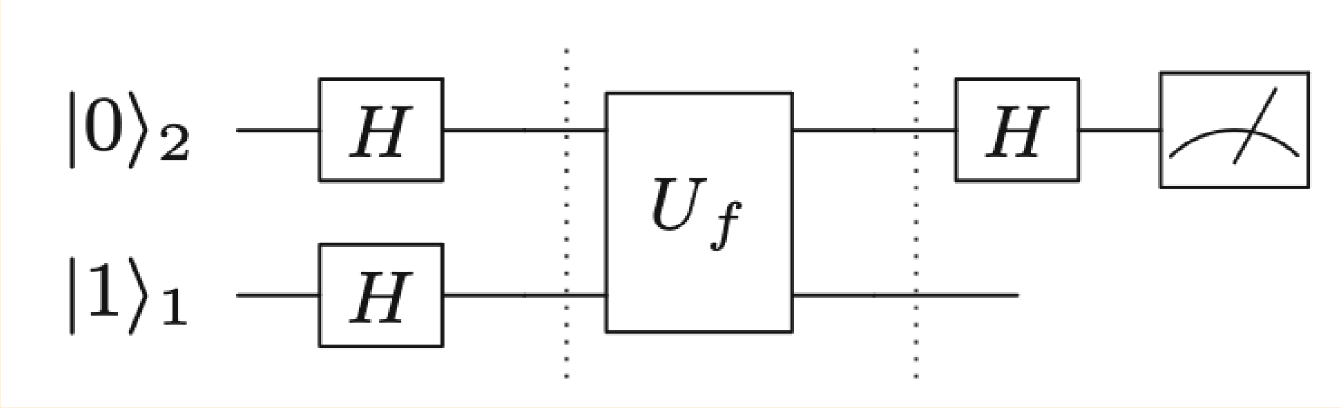
| Gate | Circuit representation | Matrix representation | Dirac representation |
|---|---|---|---|
| $X$ | $-\boxed{X}-$ | $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ | $\|1\rangle\langle 0\| + \|0\rangle\langle 1\|$ |
| $Y$ | $-\boxed{Y}-$ | $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ | $i\|1\rangle\langle 0\| - i\|0\rangle\langle 1\|$ |
| $Z$ | $-\boxed{Z}-$ | $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ | $\|1\rangle\langle 0\| - \|0\rangle\langle 1\|$ |
| $H$ | $-\boxed{H}-$ | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ | $\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)\langle 0\| + \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)\langle 1\|$ |
| $S$ | $-\boxed{S}-$ | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ | $\frac{1}{\sqrt{2}}\|0\rangle\langle 0\| + \frac{1}{\sqrt{2}}i\|1\rangle\langle 1\|$ |
| $R$ | $-\boxed{R}-$ | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & \exp(-i\pi/4) \end{pmatrix}$ | $\frac{1}{\sqrt{2}}\|0\rangle\langle 0\| + \frac{1}{\sqrt{2}}\exp^{-i\pi/4}\|1\rangle\langle 1\|$ |

—

Image from M. Schuld, F. Petruccione (2018)

# Quantum circuits

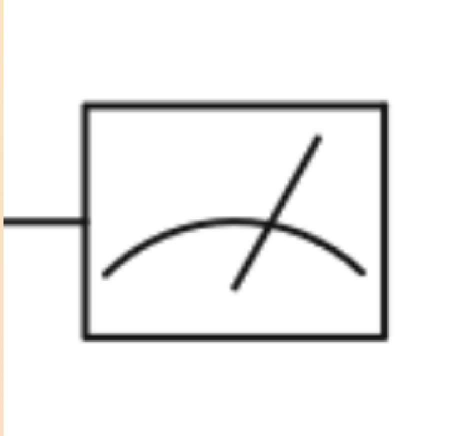
- Typically measurements are performed at the end of the circuit
- Efficient quantum algorithms: their decomposition into circuit grows at most polynomially with input size
- Popular algorithms: Grover (search a list), Shor (find the prime factors of an integer)




—
Image from M. Schuld, F. Petruccione (2018)

# Measurement

- Measuring collapses the probability distribution to one classical bit of information
- Example: Hadamard gate
  - measure $|\psi >= \alpha|0 > +\beta|1 >$
  - obtain $0$ with probability $\alpha^2$ and $1$ with probability $\beta^2$
  - after measurement, the state will be either $|0 >$ or $|1 >$
- Independent measurements of $|\psi >$ are impossible (no-cloning theorem)



—

Image from M. Schuld, F. Petruccione (2018)

# The Curse of Dimensionality

- How many samples do we need to estimate $f^\star$, depending on assumptions on its regularity?

# The Curse of Dimensionality

- How many samples do we need to estimate $f^\star$, depending on assumptions on its regularity?
- $f^\star$ constant $\longrightarrow$ need only 1 sample

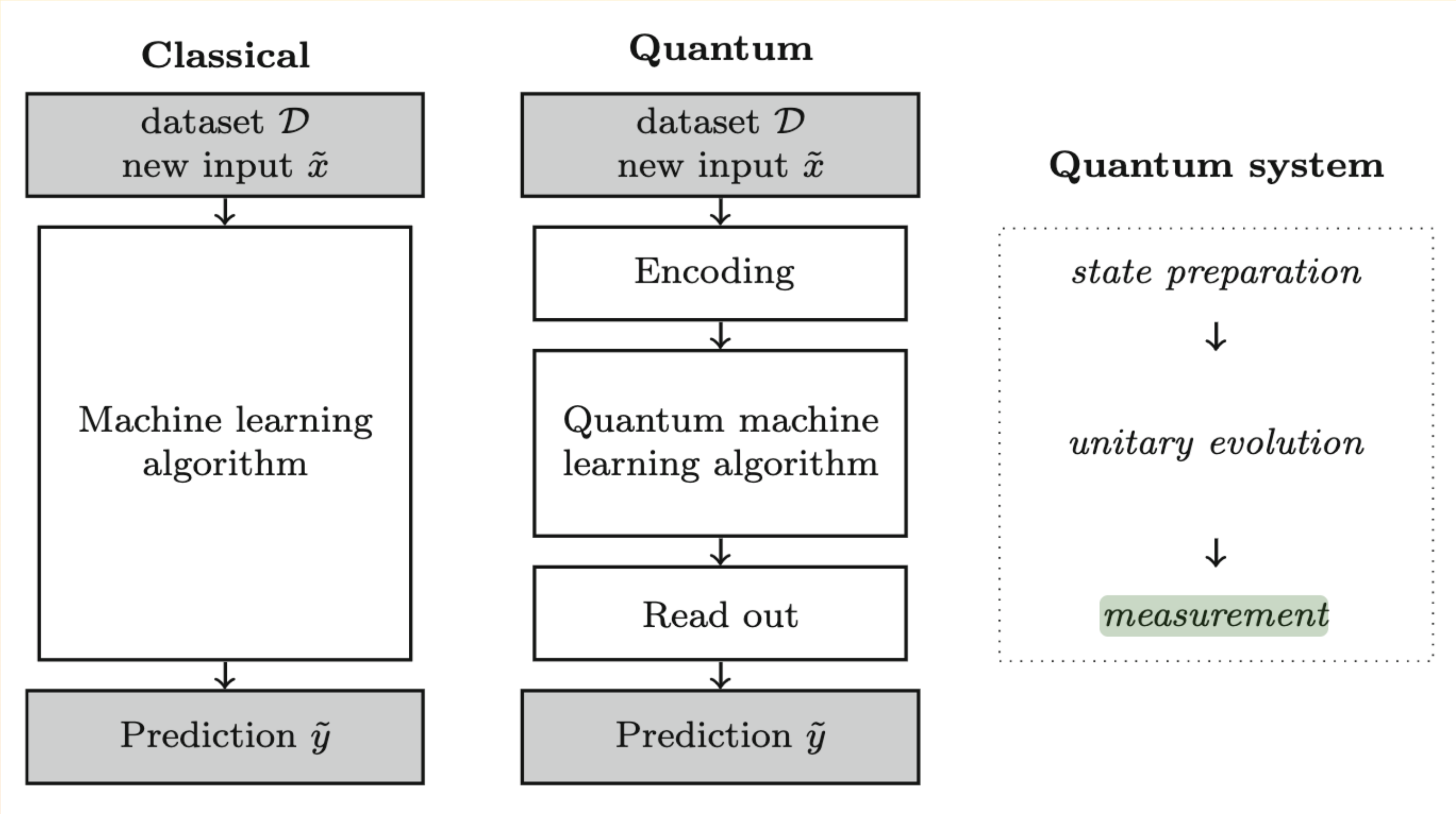
# The Curse of Dimensionality

- How many samples do we need to estimate $f^\star$, depending on assumptions on its regularity?
- $f^\star$ constant $\longrightarrow$ need only 1 sample
- $^\star$ linear $\longrightarrow$ need $d$ samples

# The Curse of Dimensionality

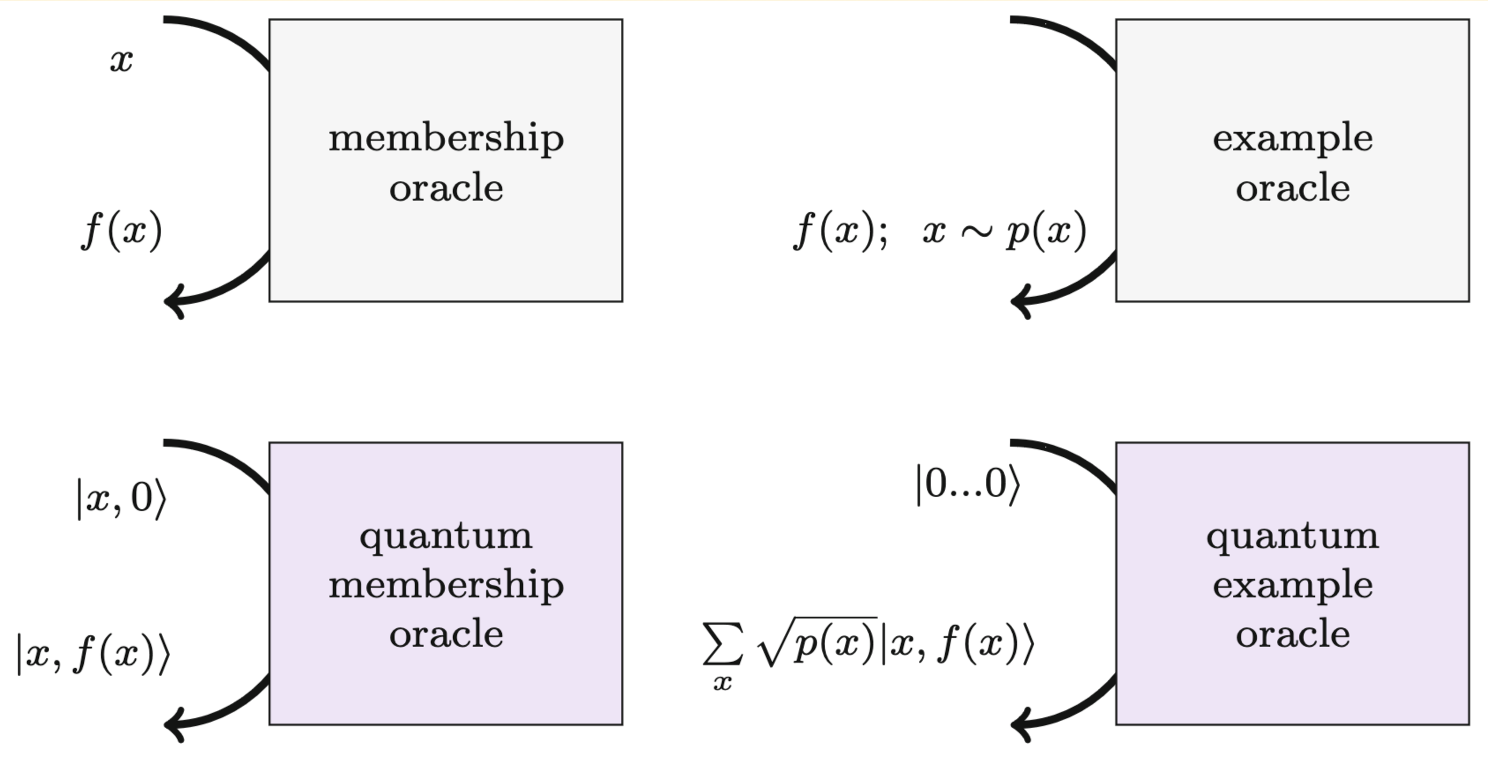
- How many samples do we need to estimate $f^\star$, depending on assumptions on its regularity?
- $f^\star$ constant $\longrightarrow$ need only 1 sample
- $^\star$ linear $\longrightarrow$ need $d$ samples
- It can be demonstrated that in $n$ dimensions you need $n \sim \epsilon^{-d}$ samples

# Classical vs Quantum ML




—
Image from M. Schuld, F. Petruccione (2018)

# Label data




—

Image from M. Schuld, F. Petruccione (2018)

# Network structure

- Qubit operations can represent rather naturally neural networks

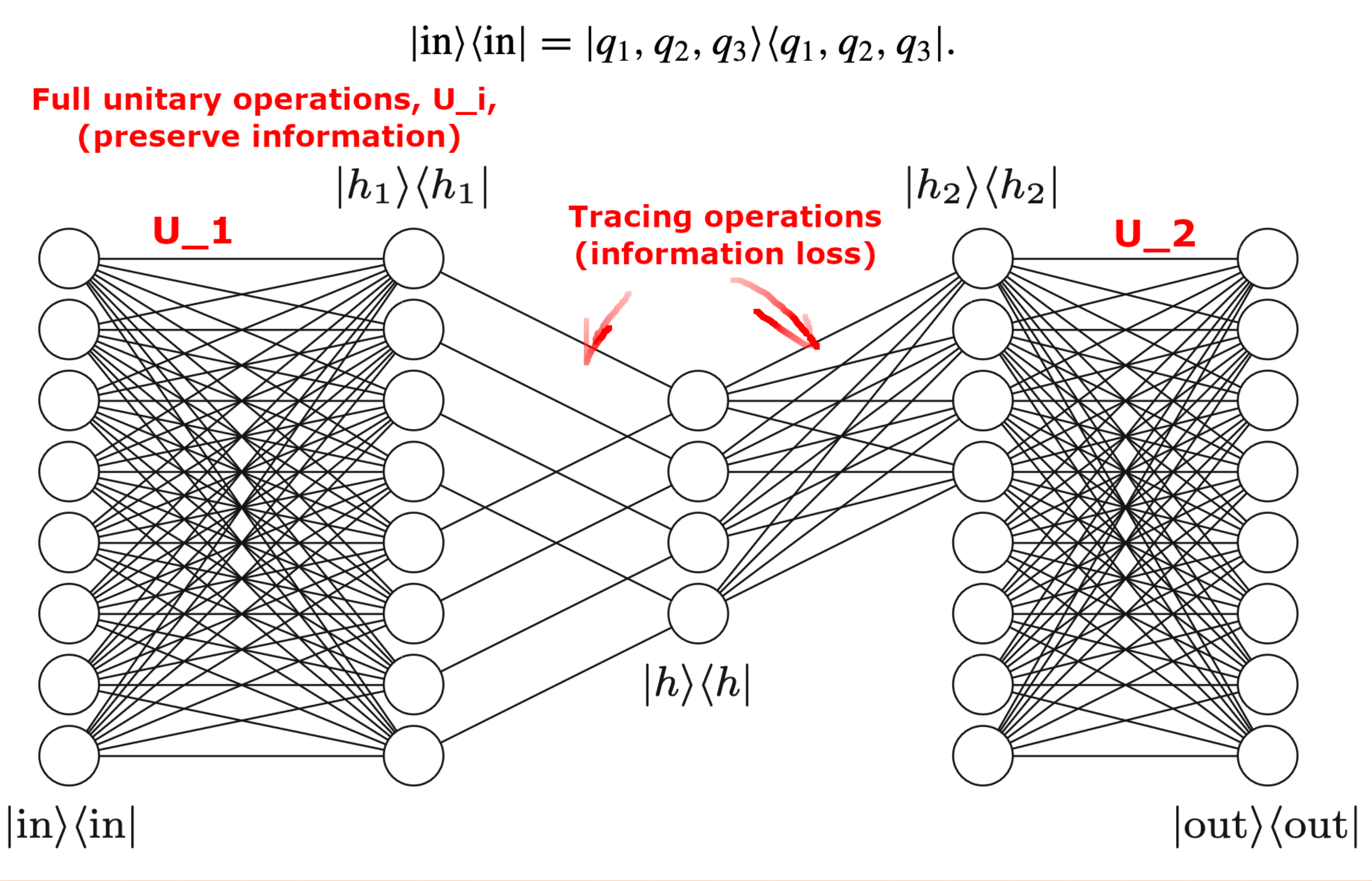



Image from M. Schuld, F. Petruccione (2018)

# Training the algorithms

- Evolution through unitary transformations emulates linear algebra
  - Can be used to invert matrices and find weights
- Perceptrons: decision boundaries as points in an hypersphere, find hyperplanes

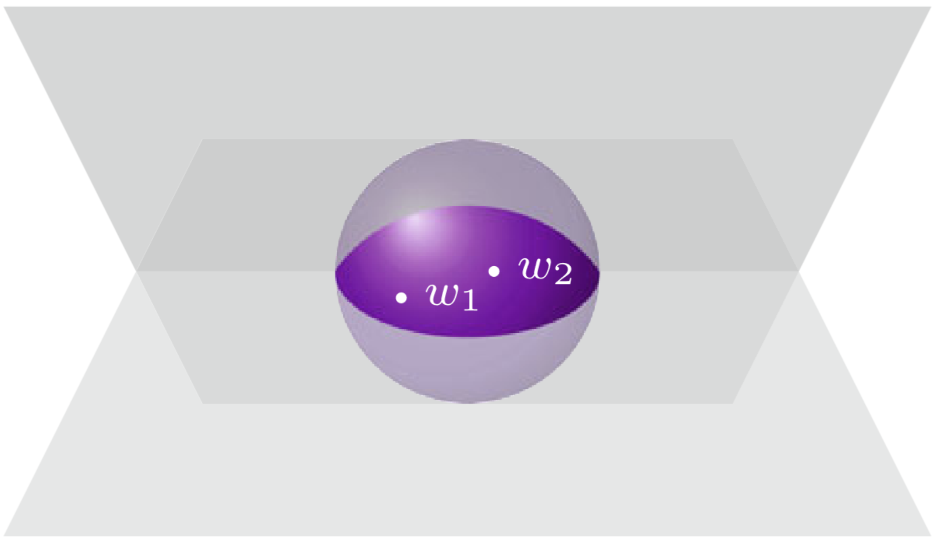



**Fig. 7.2** In the dual representation, the normal vector of the separating hyperplanes $w^1$ and $w^2$ of a perceptron model are represented by points on a hypersphere, while the training set identifies a feasible region for the separating hyperplanes. Each training vector corresponds to a plane that 'cuts away' part of the hypersphere (illustrated by the grey planes)

—
Image from M. Schuld, F. Petruccione (2018)

# Training the algorithms

- Hybrid classical/quantum training schemes also possible

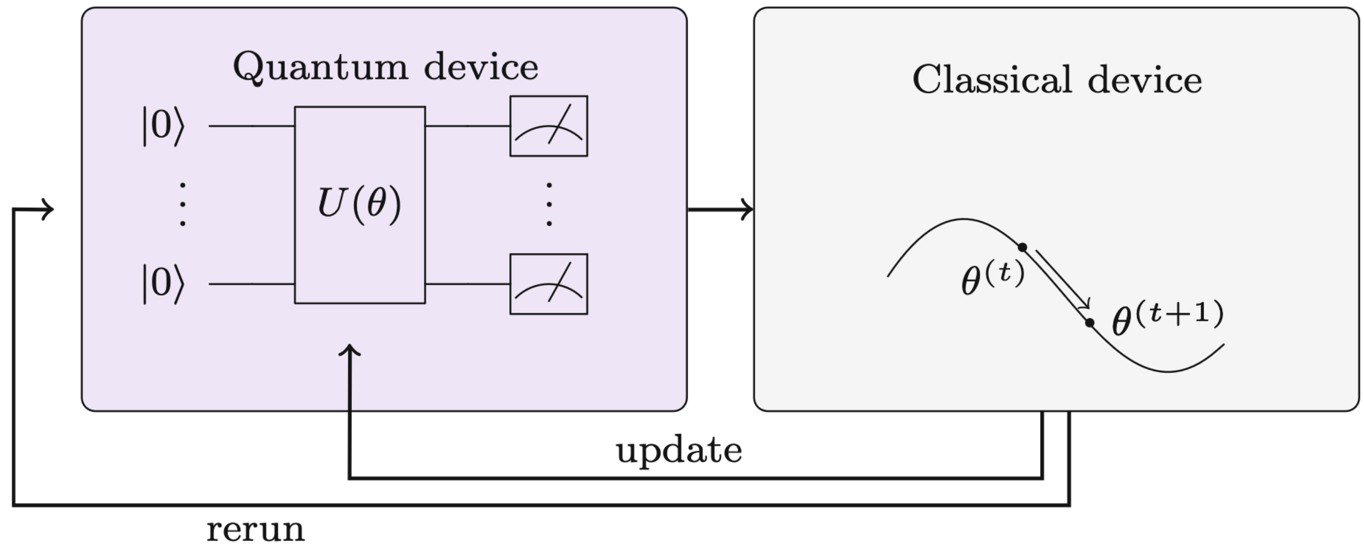



**Fig. 7.3** The idea of a hybrid quantum-classical training algorithm for variational circuits is to use the quantum device to compute terms of an objective function or its derivatives, and subsequently use a classical device to compute better circuit parameters with respect to the objective. The entire routine gets iterated until the objective is minimised or maximised

—
Image from M. Schuld, F. Petruccione (2018)

# Training the algorithms

- Gradient descent exploits intrinsic analytic differentiability of quantum circuits

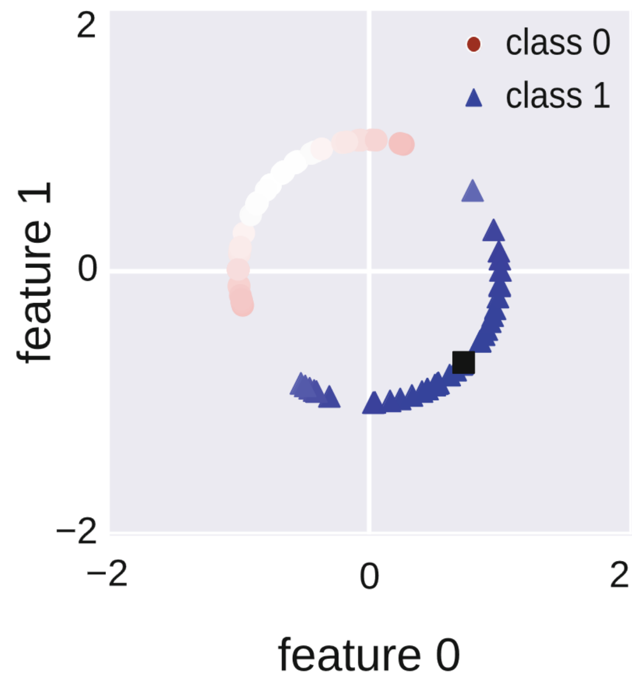
$$
\begin{aligned}
\partial_\mu \, \langle \psi(x,\theta)|\sigma_z|\psi(x,\theta)\rangle &= \langle 0| \ldots \partial_\mu \mathrm{e}^{-i\mu\sigma} \ldots \sigma_z \ldots \mathrm{e}^{i\mu\sigma} \ldots |0\rangle \\
&\quad + \langle 0| \ldots \mathrm{e}^{-i\mu\sigma} \ldots \sigma_z \ldots \partial_\mu \mathrm{e}^{i\mu\sigma} \ldots |0\rangle \\
&= \langle 0| \ldots (-i\sigma)\mathrm{e}^{-i\mu\sigma} \ldots \sigma_z \ldots \mathrm{e}^{i\mu\sigma} \ldots |0\rangle \\
&\quad + \langle 0| \ldots \mathrm{e}^{-i\mu\sigma} \ldots \sigma_z \ldots (i\sigma)\mathrm{e}^{i\mu\sigma} \ldots |0\rangle \\
&= \langle 0| \ldots (1 - i\sigma)\mathrm{e}^{-i\mu\sigma} \ldots \sigma_z \ldots (1 + i\sigma)\mathrm{e}^{i\mu\sigma} \ldots |0\rangle \\
&\quad + \langle 0| \ldots (1 + i\sigma)\mathrm{e}^{-i\mu\sigma} \ldots \sigma_z \ldots (1 - i\sigma)\mathrm{e}^{i\mu\sigma} \ldots |0\rangle
\end{aligned}
$$

—
Image from M. Schuld, F. Petruccione (2018)

# Classifying new data

- Often by proximity after representing data in a unit circle

**Fig. 6.10** Example of how the kernelised binary classifier in this section weighs neighbouring training inputs to come to a decision. The inputs are normalised and lie on a unit circle. The influence of a training input on the prediction of the new input (black square) is depicted by the colour scheme, and lighter dots have less influence



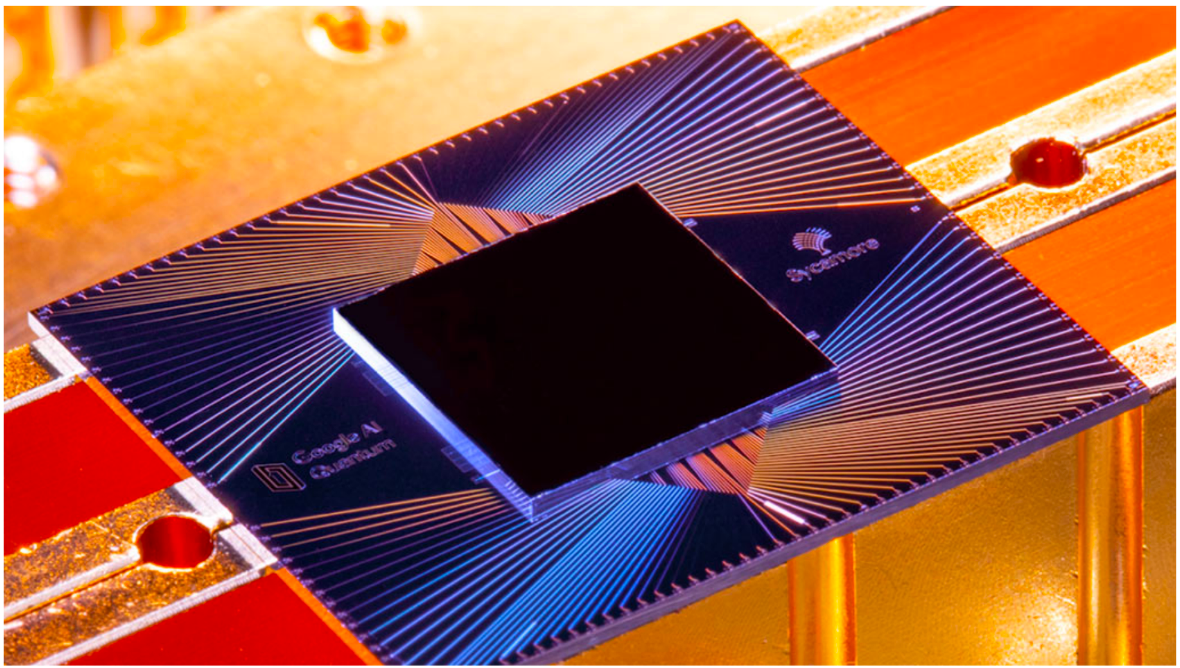
# Quantum advantage

- Measuring all information of a quantum state is subject to curse of dimensionality
- But the quantum computer has immediate access to all this information and can produce the result
  - For instance, yes/no decision
- Exponential speedup by design!!!

# Quantum Supremacy?

## Google officially lays claim to quantum supremacy

A quantum computer reportedly beat the most powerful supercomputers at one type of calculation

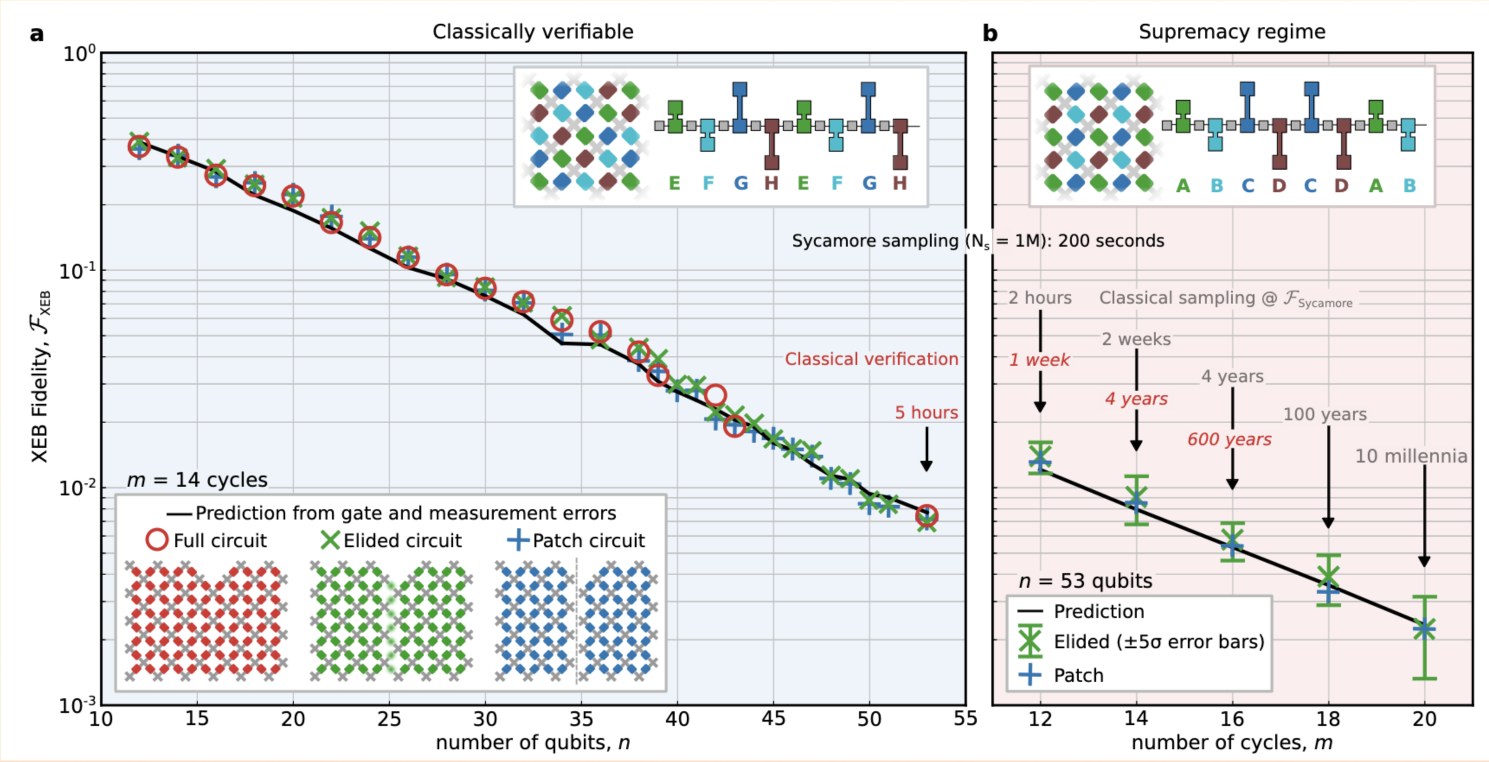



Google researchers report that their quantum computer, Sycamore, has performed a calculation that can't be achieved with any classical computer. The quantum chip (shown) must be cooled to near absolute zero to function.

—
Article source and illustration from sciencenews.org

# Quantum Supremacy?

- "How large a QC should be such that it is not possible to simulate it classically?"
  - Benchmarks for quantum supremacy are an open field of research



# Build a quantum computer


| | Superconducting loops | Trapped ions | Silicon quantum dots | Topological qubits | Diamond vacancies |
|---|---|---|---|---|---|
| **Company support** | Google, IBM, Quantum Circuits | ionQ | Intel | Microsoft, Bell Labs | Quantum Diamond Technologies |
| **Pros** | Fast working. Build on existing semiconductor industry. | Very stable. Highest achieved gate fidelities. | Stable. Build on existing semiconductor industry. | Greatly reduce errors. | Can operate at room temperature. |
| **Cons** | Collapse easily and must be kept cold. | Slow operation. Many lasers are needed. | Only a few entangled. Must be kept cold. | Existence not yet confirmed. | Difficult to entangle. |

Image credits: Graphic by C. Bickle/Science data by Gabriel Popkin

—
Figure from https://indico.cern.ch/event/970909/

# NISQ

- Noisy Intermediate-Scale Quantum computers
    - Noisy
    - 50-100 qubits only
    - Not fully-connected
- They may still be able to surpass digital computers, but noise limiting factor for size of reliable circuits

## Quantum Computing in the NISQ era and beyond

John Preskill

Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena CA 91125, USA

30 July 2018

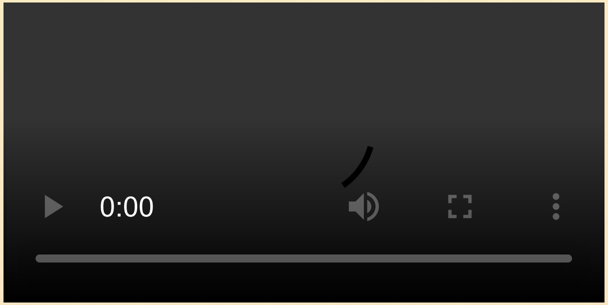
Noisy Intermediate-Scale Quantum (NISQ) technology will be available in the near future. Quantum computers with 50-100 qubits may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. NISQ devices will be useful tools for exploring many-body quantum physics, and may have other useful applications, but the 100-qubit quantum computer will not change the world right away — we should regard it as a significant step toward the more powerful quantum technologies of the future. Quantum technologists should continue to strive for more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

—

Figure from 1801.00862

# Quantum Computer ASMR

- $15mK$ temperatures
    - Essentially, listen to the sound of the refrigerator

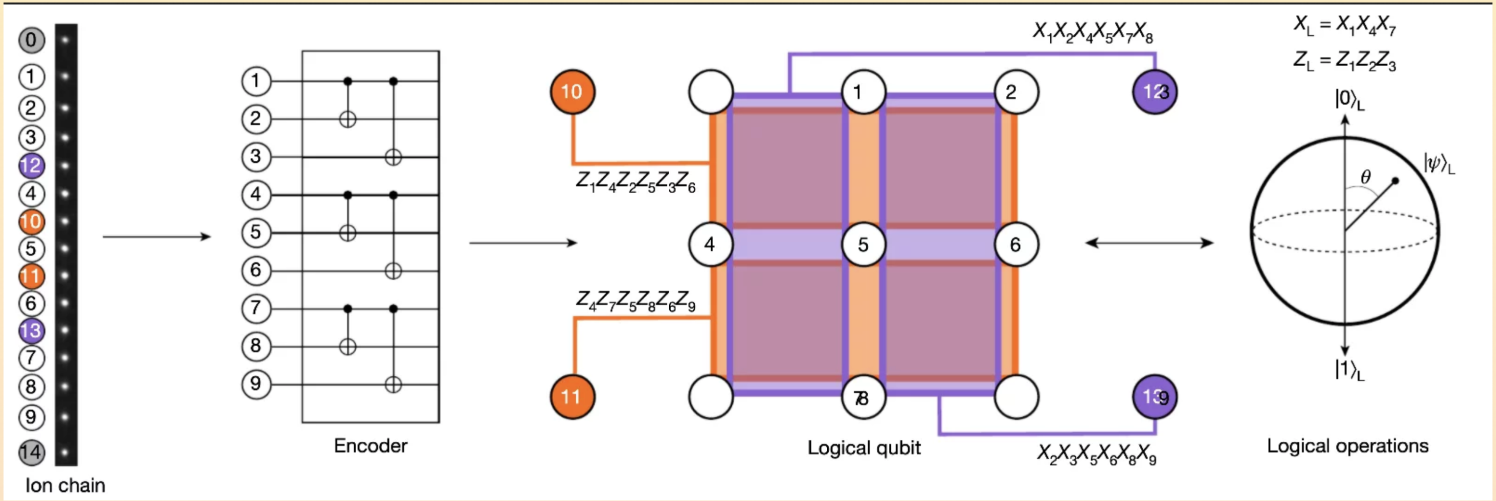



—

Full video at https://www.youtube.com/watch?v=o-FyH2A7Ed0

# Noise control

- Logical qubit constituted by a set of physical qubits
  - The additional physical qubits are tasked with error correction



# Quantum supremacy revisited

- Proofs or empirical tests to check for quantum supremacy
  - Further reading: 10.1038/s41534-017-0018-2

## Classical

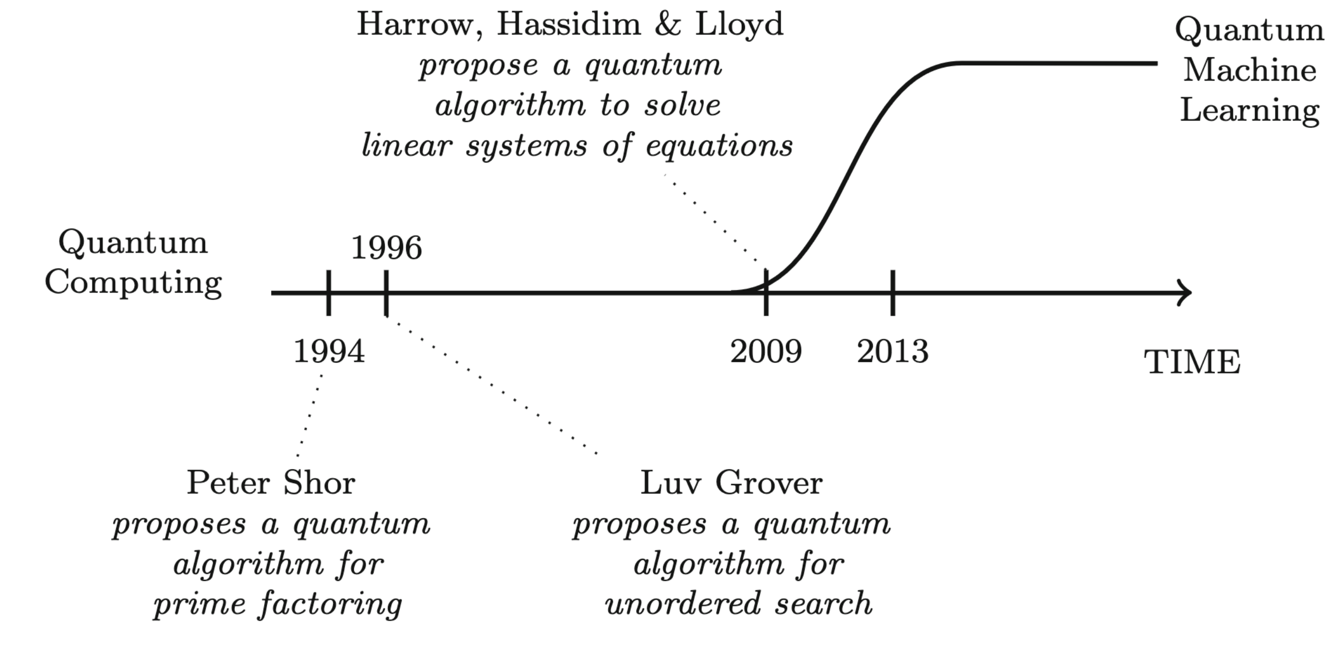



**Fig. 3.3** Timeline of quantum computing and quantum machine learning

—

Image from M. Schuld, F. Petruccione (2018)

# Speedup holy grail

- Provable quantum speedup
- Strong quantum speedup
- Common quantum speedup
- Potential quantum speeedup
- Limited quantum speedup

Holy grail: provable exponential speedup

# Speedup in QBLAS

- Quantum Basic Linear Algebra subroutines
  - Quantum Fourier Transform
  - Quantum Phase Estimation
  - HHL (quantum solver for linear systems of equations)
- Some QML algorithms rely on exponential speedup provided by QBLAS
- Common issues: load input, read output, decide circuit size

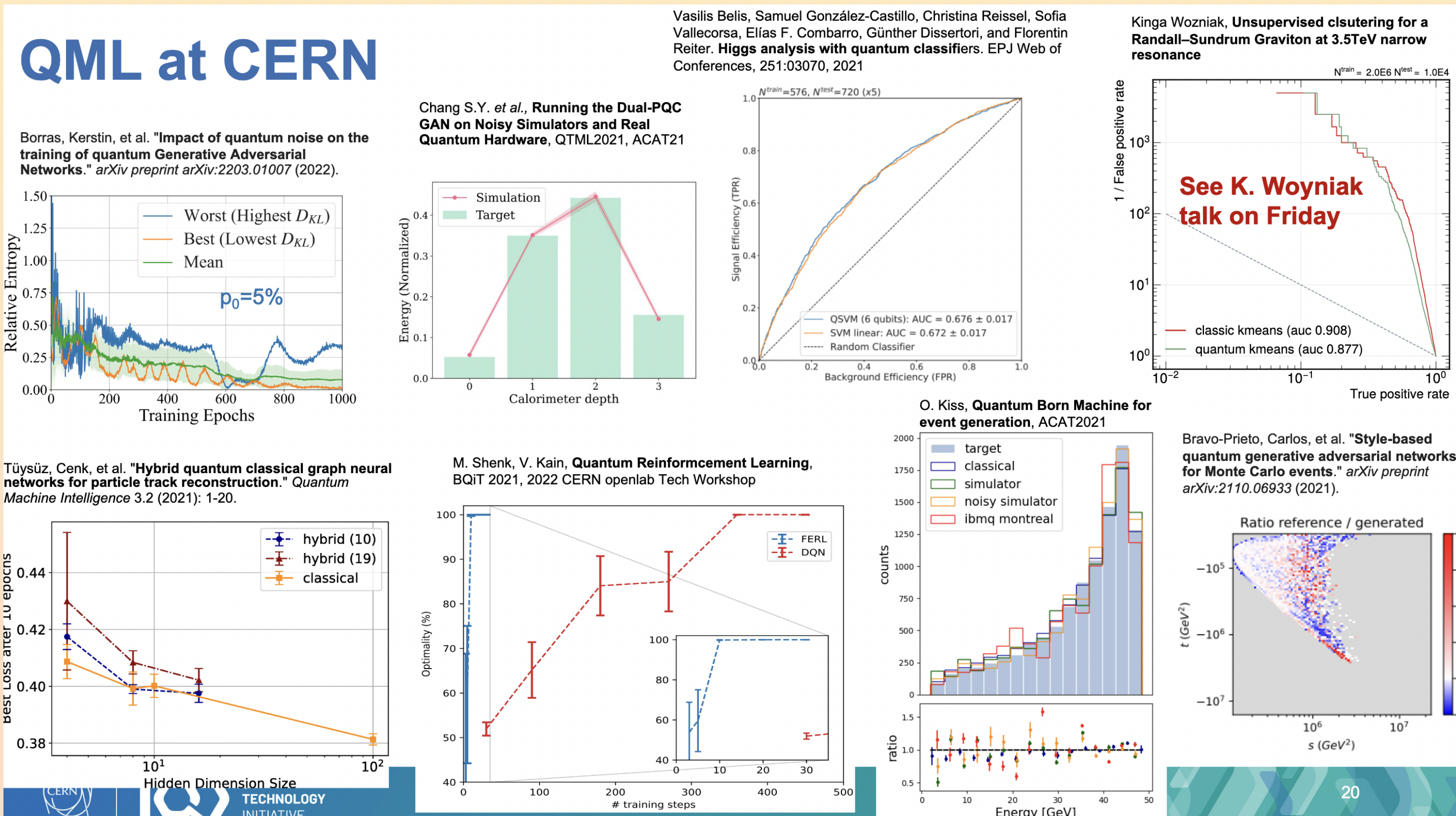
# Open questions

- Does learning from quantum data produce different results than from classical data?
- How can we combine data genereration and analysis effectively?
- Can we design algorithms to solve otherwise intractable problems?
- ...

- Don't automatically trust anyone who promises to have successfully treated a quantum problem

# QML in fundamental research

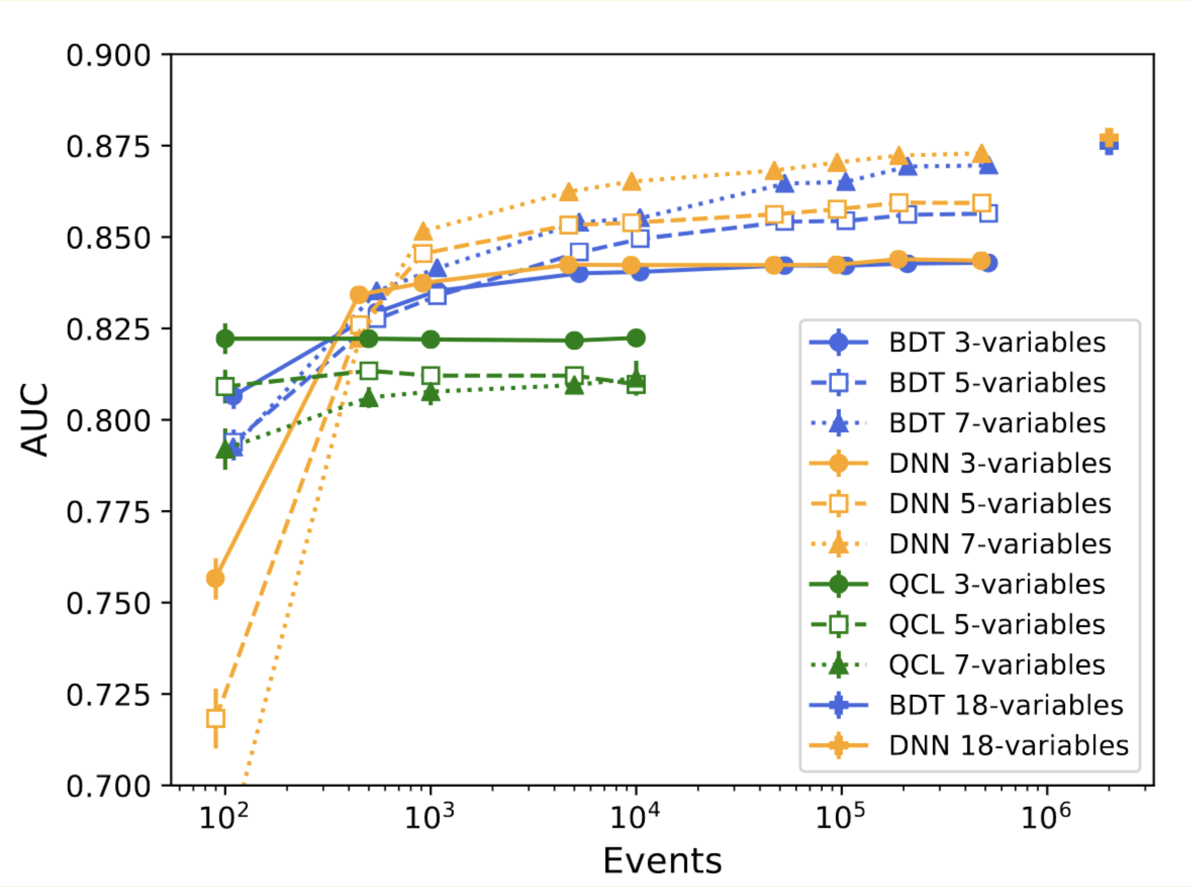
- So far, mostly as good as classical methods
- Must identify use cases where a quantum approach can be more effective




—

Slide by Sofia Vallecorsa

# Same performance, fewer samples




—
Image from 2002.09935

# How can I work on Quantum ML?

- Several implementations (mostly in python), hiding away the raw calculations
  - Pennylane
  - Tensorflow Quantum
  - Qiskit
  - ...

```
import pennylane as qml


def QSP_circ(phi, W):
    """This circuit applies the SPO. The components in the matrix
    representation of the final unitary are polynomials!
    """
    qml.Hadamard(wires=0)  # set initial state |+>
    for angle in phi[:-1]:
        qml.RZ(angle, wires=0)
        qml.QubitUnitary(W, wires=0)

    qml.RZ(phi[-1], wires=0)  # final rotation
    qml.Hadamard(wires=0)  # change of basis |+> , |->
    return
```

—

Image from Pennylane demos

# Summary

- Neural networks rely on automatic differentiation for gradient descent
- Digital networks
  - Accelerators: software and hardware tricks in common hardware
- Spiking networks
  - Encode signals in neuron spikes
  - Event-based processign: natural time and energy advantage
  - Deploy in CMOS or memristors
- Quantum Networks
  - Machine Learning based in quantomechanical properties (interference effects)
  - Encode information in a richer structure (qubit)
  - Quantum supremacy: expected exponential speedup in many problems

# Tomorrow the workshop begins at 9 AM

# Thanks for attending!!!

**I hope to have provided some base pointers for you to then go more into detail on these topics**

# Thank you!

*Welcome back to regular Galician weather* 😜

—
Photo by me!