Quantum Technology and Artificial Intelligence

Morten Hjorth-Jensen

Department of Physics and Center for Computing in Science Education, University of Oslo, Norway

Dscience seminar, UiO, April 3, 2025

What is this talk about?

The main emphasis is to give you a short and hopefully pedestrian introduction to the whys and hows of machine learning and quantum technologies. And why this could (or should) be of interest.

Thanks to many

Jane Kim (MSU), Julie Butler (MSU), Patrick Cook (MSU), Danny Jammooa (MSU), Daniel Bazin (MSU), Dean Lee (MSU), Witek Nazarewicz (MSU), Michelle Kuchera (Davidson College), Even Nordhagen (UiO), Robert Solli (UiO, Expert Analytics), Bryce Fore (ANL), Alessandro Lovato (ANL), Stefano Gandolfi (LANL), Francesco Pederiva (UniTN), and Giuseppe Carleo (EPFL). Niyaz Beysengulov and Johannes Pollanen (experiment, MSU); Zachary Stewart, Jared Weidman, and Angela Wilson (quantum chemistry, MSU) Jonas Flaten, Oskar, Leinonen, Øyvind Sigmundson Schøyen, Stian Dysthe Bilek, and Håkon Emil Kristiansen (UiO). Excuses to those I have omitted.

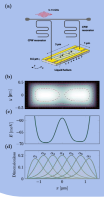
And sponsors

1. National Science Foundation, US (various grants)
2. Department of Energy, US (various grants)
3. Research Council of Norway (various grants) and my employers University of Oslo (1999-present) and Michigan State University (former, 2012-2024)

Quantum technology and machine learning/AI

Quantum Hardware

- Quantum control and sensors
- Quantum computers
- Catalysis



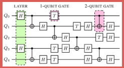
AI for Quantum Technology

- Quantum mechanical many-body theories
- Molecular dynamics
- Density functional theory

Quantum Technology and AI

Quantum Algorithms for AI

- Quantum computing
- Quantum algorithms
- Quantum machine learning
- Quantum inspired classical computing



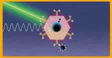
Societal impact and products

New medicines

Better shipping

Financial innovation

Quantum readiness



Quantum sensors and computers

- Drug discovery
- Quantum sensing
- Supply chain optimization
- Financial market modelling and risk analysis
- Material discovery

What is Machine Learning?

Machine Learning (ML) is the study of algorithms that improve through data experience.

Types of Machine Learning:

- **Supervised Learning:** Labeled data for classification or regression.
- **Unsupervised Learning:** No labels; discover hidden patterns.
- **Reinforcement Learning:** Learning through interaction with the environment.

ML Workflow:

$$\text{Data} \rightarrow \text{Model Training} \rightarrow \text{Prediction}$$

What is Quantum Computing?

Quantum computing leverages principles of quantum mechanics to perform computations beyond classical capabilities.

Key Concepts:

- **Superposition:** Qubits can exist in a combination of states.
- **Entanglement:** Correlation between qubits regardless of distance.
- **Quantum Interference:** Probability amplitudes interfere to solve problems.

Qubit Representation:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad |\alpha|^2 + |\beta|^2 = 1$$

What is Quantum Machine Learning?

Quantum Machine Learning (QML) integrates quantum computing with machine learning algorithms to exploit quantum advantages.

Motivation:

- High-dimensional Hilbert spaces for better feature representation.
- Quantum parallelism for faster computation.
- Quantum entanglement for richer data encoding.

Quantum Speedups in ML

Why Quantum?

- **Quantum Parallelism:** Process multiple states simultaneously.
- **Quantum Entanglement:** Correlated states for richer information.
- **Quantum Interference:** Constructive and destructive interference to enhance solutions.

Example - Grover's Algorithm:

$$\text{Quantum Search Complexity: } O(\sqrt{N}) \text{ vs. } O(N)$$

Advantage: - Speedups in high-dimensional optimization and linear algebra problems.

Challenges and Limitations

1. **Quantum Hardware Limitations:**
   - Noisy Intermediate-Scale Quantum (NISQ) devices.
   - Decoherence and limited qubit coherence times.
2. **Data Encoding:**
   - Efficient embedding of classical data into quantum states.
3. **Scalability:**
   - Difficult to scale circuits to large datasets.

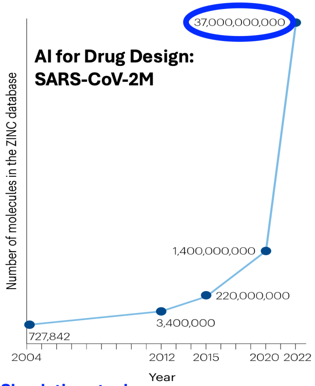
AI/ML and some statements you may have heard (and what do they mean?)

1. Fei-Fei Li on ImageNet: **map out the entire world of objects** (The data that transformed AI research)
2. Russell and Norvig in their popular textbook: **relevant to any intellectual task; it is truly a universal field** (Artificial Intelligence, A modern approach)
3. Woody Bledsoe puts it more bluntly: **in the long run, AI is the only science** (quoted in Pamilla McCorduck, Machines who think)

If you wish to have a critical read on AI/ML from a societal point of view, see Kate Crawford's recent text Atlas of AI.

Here: with AI/ML we intend a collection of machine learning methods with an emphasis on statistical learning and data analysis

Machine learning and AI models are computationally expensive



Simulations took:

90 days on **250 GPUs & 640 cpu-core**
1 day on **27000 GPU** Summit supercomputer

Nature Reviews Drug Discovery **23**, 141–155 (2024)

TOP4 Powerful Supercomputers as of Nov 2024

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581
		76	1,353.00	2,055.72	24,607
	Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States				
4	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	

Advance nuclear weapon science and scientific discovery.

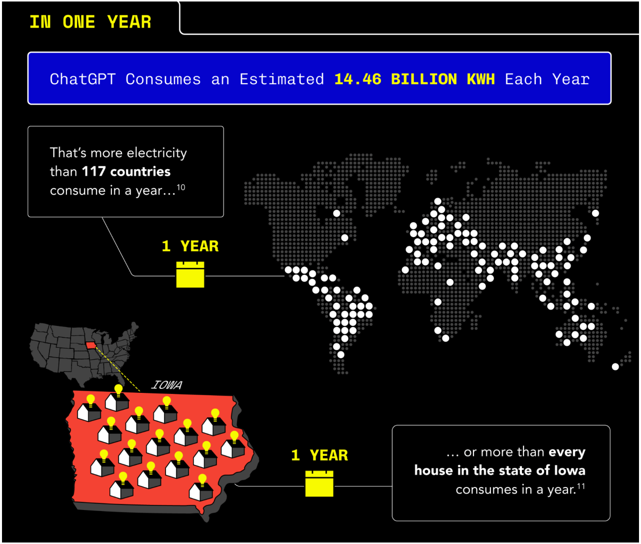
US$600 million

nature reviews drug discovery

Perspective | Published: 08 December 2023

Integrating QSAR modelling and deep learning in drug discovery: the emergence of deep QSAR

And power greedy, perhaps quantum computers can reduce the impact?



Taken from https://www.businessenergyuk.com/knowledge-hub/

Main categories of Machine Learning

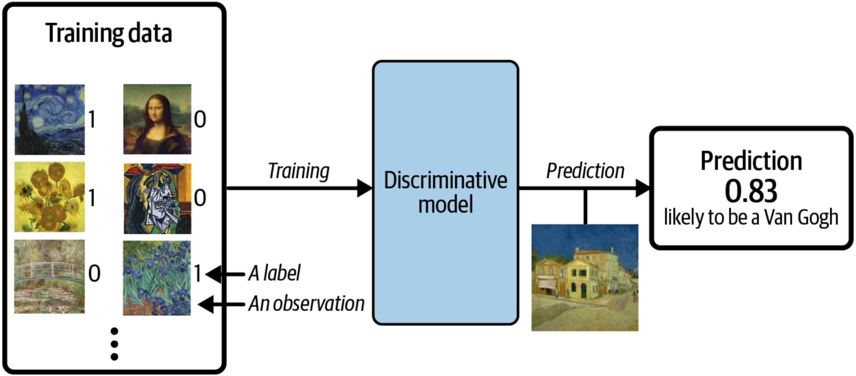
Another way to categorize machine learning tasks is to consider the desired output of a system. Some of the most common tasks are:

- Classification: Outputs are divided into two or more classes. The goal is to produce a model that assigns inputs into one of these classes. An example is to identify digits based on pictures of hand-written ones. Classification is typically supervised learning.
- Regression: Finding a functional relationship between an input data set and a reference data set. The goal is to construct a function that maps input data to continuous output values.
- Clustering: Data are divided into groups with certain common traits, without knowing the different groups beforehand. It is thus a form of unsupervised learning.

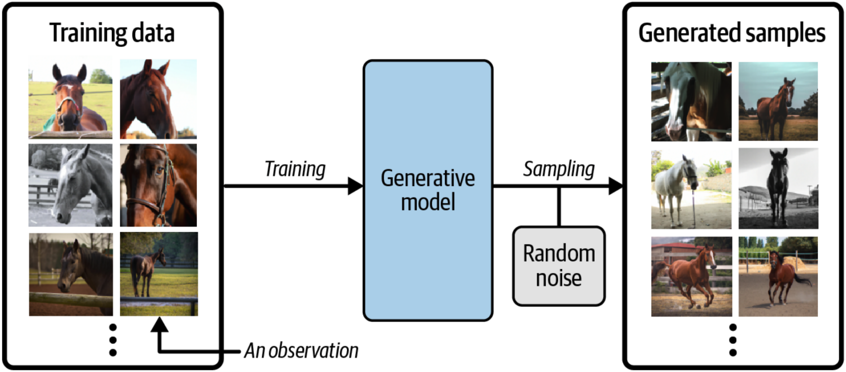
The plethora of machine learning algorithms/methods

1. Deep learning: Neural Networks (NN), Convolutional NN, Recurrent NN, Boltzmann machines, autoencoders and variational autoencoders and generative adversarial networks, stable diffusion and many more generative models
2. Bayesian statistics and Bayesian Machine Learning, Bayesian experimental design, Bayesian Regression models, Bayesian neural networks, Gaussian processes and much more
3. Dimensionality reduction (Principal component analysis), Clustering Methods and more
4. Ensemble Methods, Random forests, bagging and voting methods, gradient boosting approaches
5. Linear and logistic regression, Kernel methods, support vector machines and more
6. Reinforcement Learning; Transfer Learning and more

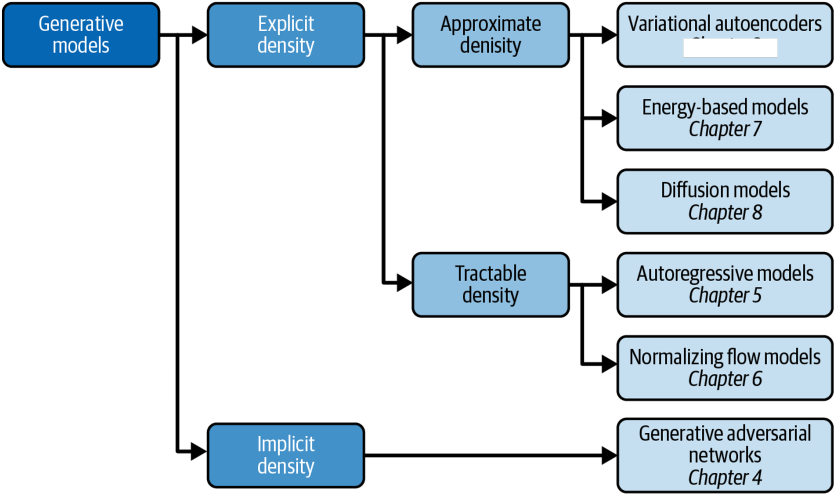
Example of discriminative modeling, taken from Generative Deep Learning by David Foster



Example of generative modeling, taken from Generative Deep Learning by David Foster



Taxonomy of generative deep learning, taken from Generative Deep Learning by David Foster



What are the basic Machine Learning ingredients?

Almost every problem in ML and data science starts with the same ingredients:

- The dataset $\boldsymbol{x}$ (could be some observable quantity of the system we are studying)
- A model which is a function of a set of parameters $\boldsymbol{\alpha}$ that relates to the dataset, say a likelihood function $p(\boldsymbol{x}|\boldsymbol{\alpha})$ or just a simple model $f(\boldsymbol{\alpha})$
- A so-called **loss/cost/risk** function $\mathcal{C}(\boldsymbol{x}, f(\boldsymbol{\alpha}))$ which allows us to decide how well our model represents the dataset.

We seek to minimize the function $\mathcal{C}(\boldsymbol{x}, f(\boldsymbol{\alpha}))$ by finding the parameter values which minimize $\mathcal{C}$. This leads to various minimization algorithms. It may surprise many, but at the heart of all machine learning algortihms there is an optimization problem.

Low-level machine learning, the family of ordinary least squares methods

Our data which we want to apply a machine learning method on, consist of a set of inputs $\boldsymbol{x}^T = [x_0, x_1, x_2, \dots, x_{n-1}]$ and the outputs we want to model $\boldsymbol{y}^T = [y_0, y_1, y_2, \dots, y_{n-1}]$. We assume that the output data can be represented (for a regression case) by a continuous function f through

$$\boldsymbol{y} = f(\boldsymbol{x}) + \epsilon.$$

Setting up the equations

In linear regression we approximate the unknown function with another continuous function $\tilde{\boldsymbol{y}}(\boldsymbol{x})$ which depends linearly on some unknown parameters $\boldsymbol{\theta}^T = [\theta_0, \theta_1, \theta_2, \dots, \theta_{p-1}]$.
The input data can be organized in terms of a so-called design matrix with an approximating function $\tilde{\boldsymbol{y}}$

$$\tilde{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\theta},$$

The objective/cost/loss function

The simplest approach is the mean squared error

$$C(\boldsymbol{\Theta}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2 = \frac{1}{n}\left\{(\boldsymbol{y} - \tilde{\boldsymbol{y}})^T(\boldsymbol{y} - \tilde{\boldsymbol{y}})\right\},$$

or using the matrix $\boldsymbol{X}$ and in a more compact matrix-vector notation as

$$C(\boldsymbol{\Theta}) = \frac{1}{n}\left\{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})\right\}.$$

This function represents one of many possible ways to define the so-called cost function.

Training solution

Optimizing with respect to the unknown parameters θ_j we get

$$\boldsymbol{X}^T \boldsymbol{y} = \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{\theta},$$

and if the matrix $\boldsymbol{X}^T \boldsymbol{X}$ is invertible we have the optimal values

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y}.$$

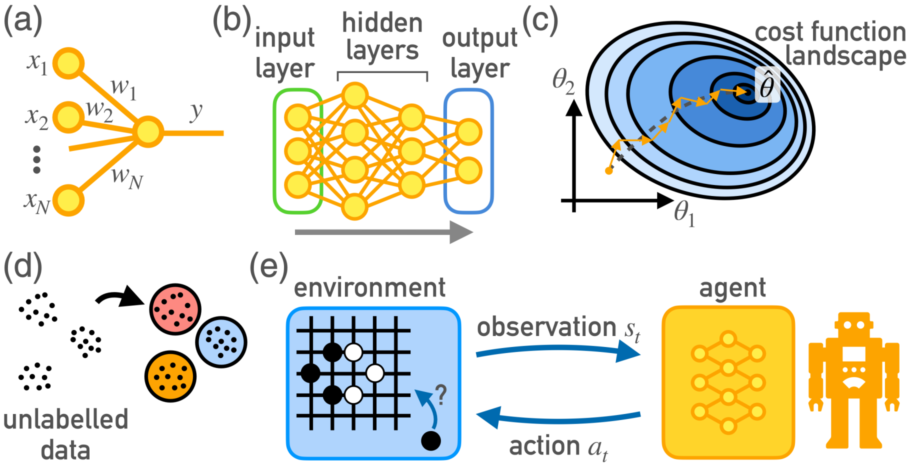
We say we 'learn' the unknown parameters $\boldsymbol{\theta}$ from the last equation.

Why Neural Networks and deep learning?

According to the *Universal approximation theorem*, a feed-forward neural network with just a single hidden layer containing a finite number of neurons can approximate a continuous multidimensional function to arbitrary accuracy, assuming the activation function for the hidden layer is a **non-constant, bounded and monotonically-increasing continuous function**.

You can think of a neural network as a universal approximator

Schematic view on Machine Learning approaches



Scientific Machine Learning

An important and emerging field is what has been dubbed as scientific ML, see the article by Deiana et al, Applications and Techniques for Fast Machine Learning in Science, Big Data **5**, 787421 (2022) `https://doi.org/10.3389/fdata.2022.787421`

The authors discuss applications and techniques for fast machine learning (ML) in science – the concept of integrating power ML methods into the real-time experimental data processing loop to accelerate scientific discovery. The report covers three main areas

1. applications for fast ML across a number of scientific domains;
2. techniques for training and implementing performant and resource-efficient ML algorithms;
3. and computing architectures, platforms, and technologies for deploying these algorithms.

ML for detectors

ELSEVIER

Engineering

Volume 6, Issue 3, March 2020, Pages 264-274

Research Artificial Intelligence—Review

A Survey of Accelerator Architectures for Deep Neural Networks

Yiran Chen [a], Yuan Xie [b], Linghao Song [a], Fan Chen [a], Tianqi Tang [b]

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.eng.2020.01.007

Physics driven Machine Learning

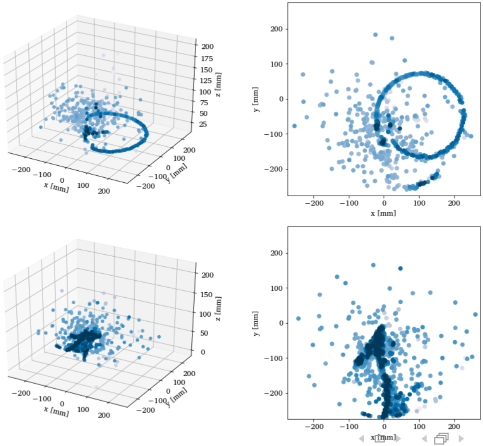
Another hot topic is what has loosely been dubbed **Physics-driven deep learning**. See the recent work on Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nature Machine Learning, vol 3, 218 (2021).

And more

- An important application of AI/ML methods is to improve the estimation of bias or uncertainty due to the introduction of or lack of physical constraints in various theoretical models.
- We expect to use AI/ML algorithms and methods to improve our knowledge about correlations of physical model parameters in data for complex systems. Deep learning methods show great promise in circumventing the exploding dimensionalities encountered in many problems.

Argon-46 by Solli et al., NIMA 1010, 165461 (2021)

Each row is one event in two projections, where the color intensity of each point indicates higher charge values recorded by the detector. The bottom row illustrates a carbon event with a large fraction of noise, while the top row shows a proton event almost free of noise.



Many-body physics, Quantum Monte Carlo and deep learning

Given a hamiltonian H and a trial wave function Ψ_T, the variational principle states that the expectation value of $\langle H\rangle$, defined through

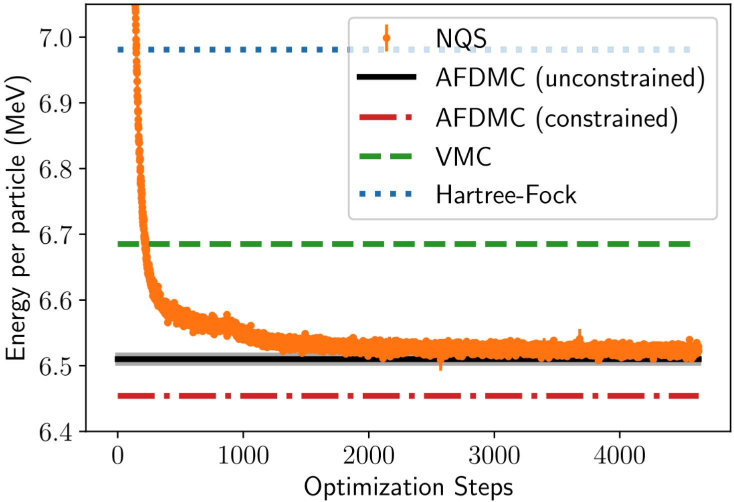
$$\langle E\rangle = \frac{\int d\boldsymbol{R}\Psi^*_T(\boldsymbol{R})H(\boldsymbol{R})\Psi_T(\boldsymbol{R})}{\int d\boldsymbol{R}\Psi^*_T(\boldsymbol{R})\Psi_T(\boldsymbol{R})},$$

is an upper bound to the ground state energy E_0 of the hamiltonian H, that is

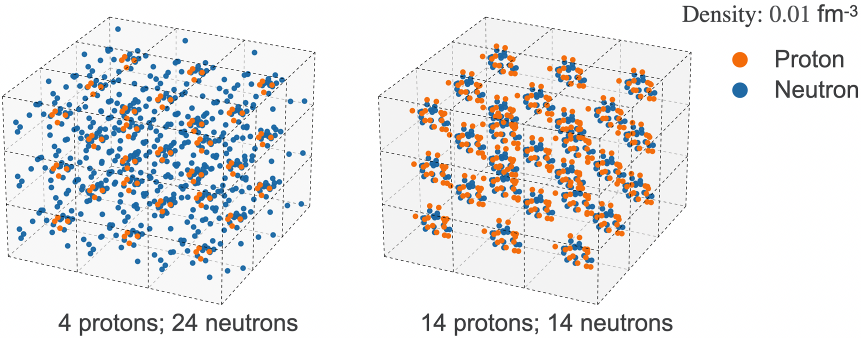
$$E_0 \le \langle E\rangle.$$

In general, the integrals involved in the calculation of various expectation values are multi-dimensional ones. Traditional integration methods such as the Gauss-Legendre will not be adequate for say the computation of the energy of a many-body system. **Basic philosophy: Let a neural network find the optimal wave function**

Dilute neutron star matter from neural-network quantum states by Fore *et al.*, Physical Review Research 5, 033062 (2023) at density $\rho = 0.04\ \mathrm{fm}^{-3}$

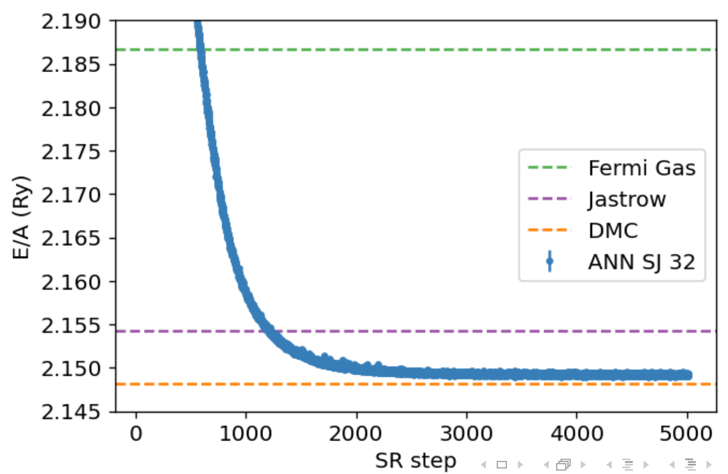


Self-emerging clustering Fore *et al.*, `https://www.nature.com/articles/s42005-025-02015-2`

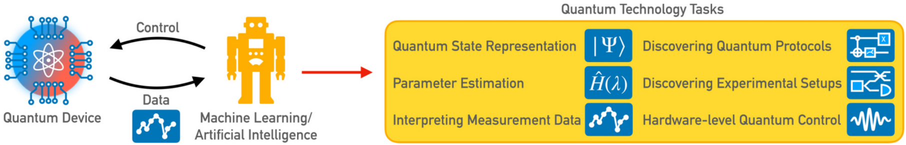


B. Fore, arXiv:2407.21207

The electron gas in three dimensions with $N = 14$ electrons (Wigner-Seitz radius $r_s = 2$ a.u.), Kim *et al.*,`https://journals.aps.org/prb/abstract/10.1103/PhysRevB.110.035108`



And then quantum engineering and ML/AI



What is Quantum Entanglement?

Quantum Entanglement is a quantum phenomenon where two or more particles become correlated in such a way that the state of one particle directly affects the state of the other, regardless of distance.

Key Features:

- Non-local correlations
- No classical analog
- Violates Bell's inequalities

Entangled State Example:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Quantum Teleportation:

- Entanglement enables the transmission of quantum states using classical communication.
- No need to send the physical quantum particle.

Advantage:

- Instantaneous state transfer within quantum mechanics constraints.
- Quantum networks rely on entanglement for secure communication.

Quantum Key Distribution:

- Entanglement ensures secure communication.
- Eavesdropping disturbs quantum states, revealing interception attempts.
- Any measurement by a third party collapses the wavefunction.
- Ensures security based on quantum mechanics, not computational hardness.

Advantage: Unconditional security guaranteed by the laws of physics.

Speedup in Quantum Algorithms:

- Entanglement provides exponential state space.
- Quantum parallelism arises from entangled qubits.

Grover's Algorithm:

$$\mathcal{O}(\sqrt{N}) \text{ vs. } \mathcal{O}(N)$$

Shor's Algorithm:

$$\text{Factoring in } \mathcal{O}((\log N)^3)$$

Quantum Metrology:

- Uses entangled states for ultra-precise measurements.
- Overcomes the classical shot-noise limit.

Heisenberg Limit:

$$\Delta\theta \geq \frac{1}{N},$$

where N is the number of entangled particles.

Advantage:

- Quantum entanglement improves sensitivity beyond classical limits.

Challenges of Quantum Entanglement

Decoherence:

- Entangled states are fragile.
- Interaction with the environment collapses the wavefunction.

Scalability:

- Difficult to entangle large numbers of qubits.
- Error correction requires complex protocols.

Measurement Problem:

- Measurement destroys entanglement.
- Trade-off between information gain and entanglement preservation.

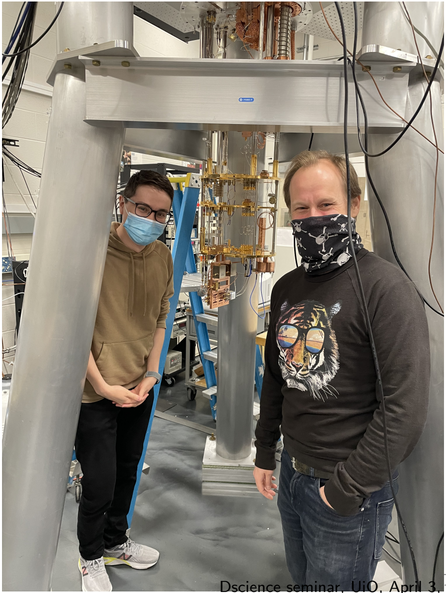
Di Vincenzo criteria

Quantum computing requirements

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

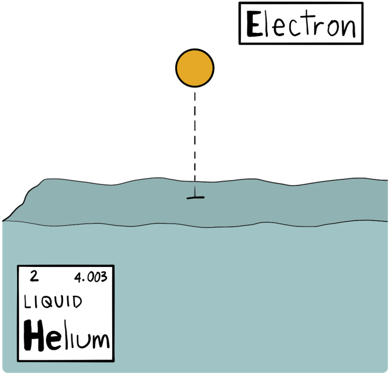
Important properties, electrons on helium

1. Long coherence times
2. Highly connect qubits
3. Many qubits in a small area
4. CMOS compatible
5. Fast gates



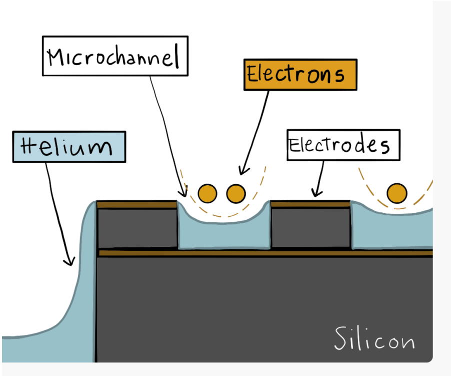
Single electrons can make great qubits

At the heart is the trapping and control of individual electrons floating above pools of superfluid helium. These electrons form the qubits of our quantum computer, and the purity of the superfluid helium protects the intrinsic quantum properties of each electron. The ultimate goal is to build a large-scale quantum computer based on quantum magnetic (spin) state of these trapped electrons.



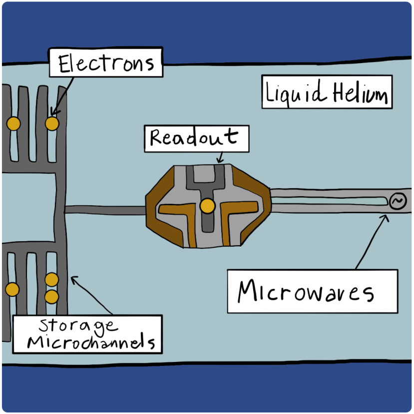
Trapping electrons in microchannels

Microchannels fabricated into silicon wafers are filled with superfluid helium and energized electrodes. Together with the natural electron trapping properties of superfluid helium, these allow for the precision trapping of individual or multiple electrons. The microchannels are only a few micrometers in size, or about five times smaller than the diameter of a human hair.



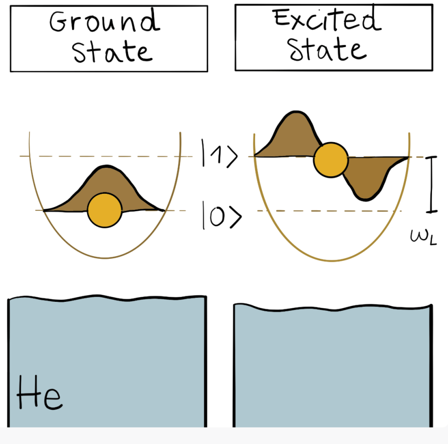
Control and readout

Microchannel regions can store thousands of electrons, from which one can be plucked and transported to the single electron control and readout area. In this region, microwave signals will interact with the electron to perform quantum logic gate operations, which will be readout via extremely fast electronics.

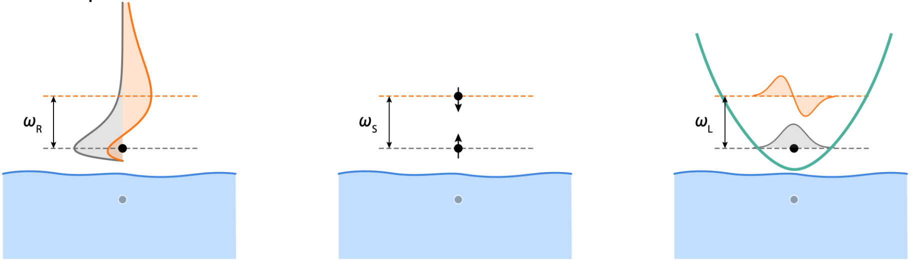


Operations for quantum computing

Quantum information can be encoded in a number of ways using single electrons. Currently, we are working with the side-to-side(lateral) quantum motion of the electron in the engineered trap. This motion can either be in its lowest energy state, the ground state, or in a number of higher-energy excited states. This electron motion also provides the readout capabilities for the ultimate goal of building a large-scale quantum computer based on the electron's magnetic moment (spin).



Qubit platforms with electrons on helium



Rydberg states	**Spin states**	**Lateral motional states**
$\omega_R/2\pi = 120$ GHz	$\omega_s/2\pi = 5$ GHz at $B = 0.2$ T $(T_2 \approx 1.5$ s$)$	$\omega_s/2\pi = 5$ GHz
P.M. Platzman and M.I. Dykman *Science* **284**(5422), pp.1967 (1999)	S. A. Lyon, *Phys. Rev. A* **74**, 052338 (2006)	D.I. Schuster et al., *Phys. Rev. Lett.* **105**, 040503 (20

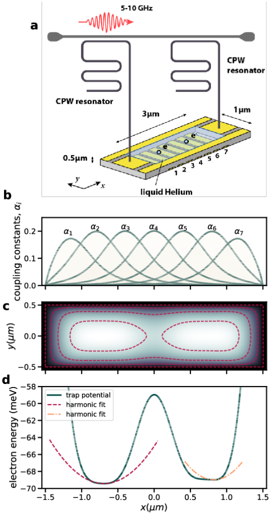
D. Konstantinov - OIST (Okinawa)
A.Chepelianskii - Universite Paris-Sud

S. Lyon - Princeton
E. Kawakami - RIKEN

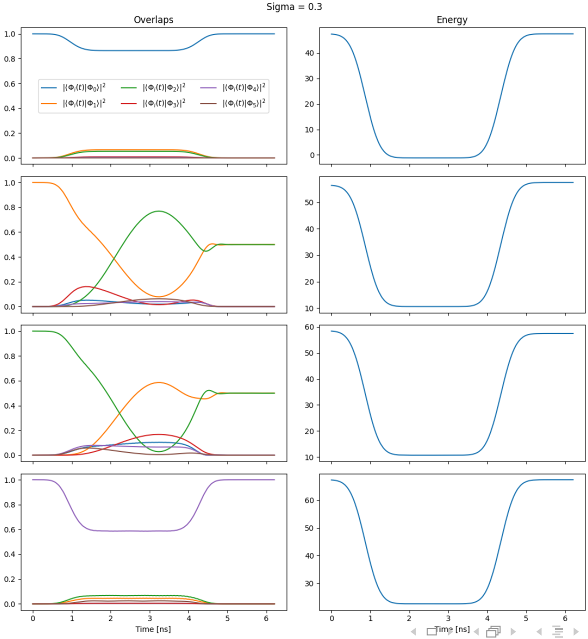
D. Schuster - University of Chicago
J. Pollanen - EeroQ/MSU

Final experimental setup

1. (a) Microdevice where two electrons are trapped in a double-well potential created by electrodes 1-7. The read-out is provided by two superconducting resonators dispersively coupled to electron's in-plane motional states.
2. (b) Coupling constants from each individual electrode beneath the helium layer.
3. (c+d) The electron's energy in a double-well electrostatic potential.



Two-qubit gates and time evolution, SWAP gate



Observations (or conclusions if you prefer)

- How do we develop insights, competences, knowledge in AI and quantum technologies that can advance a given field?
  - For example: Can we use ML to find out which correlations are relevant and thereby diminish the dimensionality problem in complex interacting many-particle systems?
  - Can we use AI/ML in detector analysis, accelerator design, analysis of experimental data and more?
  - Can we use AL/ML to carry out reliable extrapolations by using current experimental knowledge and current theoretical models?
  - How do we study entanglement in various quantum platforms? Can we use AI/ML for better design?
- The community needs to invest in relevant educational efforts and training of scientists with knowledge in AI/ML and quantum technologies
- Most likely tons of things I have forgotten

More conclusions or perspectives: Selected applications of Quantum Machine Learning

1. **Quantum mechanical many-particle systems:**
   - Simulate structures in nuclei, atoms, moleculs etc with QML.
2. **Finance:**
   - Quantum optimization for portfolio management.
3. **Image Recognition:**
   - Quantum-enhanced convolutional neural networks.

Thank you for the attention and results from references in slides)

1. Bryce Fore, Jane Kim, Morten Hjorth-Jensen, Alessandro Lovato, **Investigating the crust of neutron stars with neural-network quantum states**, Nature Communications Physics **8**, 108 (2025) and `https://www.nature.com/articles/s42005-025-02015-2`
2. Patrick Cook, Danny Jammooa, Morten Hjorth-Jensen, Daniel D. Lee, Dean Lee, **Parametric Matrix Models**, Nature Machine Learning under review and `https://arxiv.org/abs/2401.11694`
3. Niyaz R. Beysengulov, Johannes Pollanen, Øyvind S. Schøyen, Stian D. Bilek, Jonas B. Flaten, Oskar Leinonen, Håkon Emil Kristiansen, Zachary J. Stewart, Jared D. Weidman, Angela K. Wilson, Morten Hjorth-Jensen, Coulomb interaction-driven entanglement of electrons on helium, PRX Quantum 5, 030324 (2024) and `https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.5.030324`

Additional references)

1. Jane Kim, Gabriel Pescia, Bryce Fore, Jannes Nys, Giuseppe Carleo, Stefano Gandolfi, Morten Hjorth-Jensen, Alessandro Lovato, **Neural-network quantum states for ultra-cold Fermi gases**, Nature Communications Physics **7**, 148 (2024) and `https://www.nature.com/articles/s42005-024-01613-w`
2. Bryce Fore, Jane M. Kim, Giuseppe Carleo, Morten Hjorth-Jensen, Alessandro Lovato, and Maria Piarulli, **Dilute neutron star matter from neural-network quantum states**, Physical Review Research 5, 033062 (2023)
3. Robert Solli, Daniel Bazin, Michelle P. Kuchera, Ryan R. Strauss, Morten Hjorth-Jensen, *Unsupervised Learning for Identifying Events in Active Target Experiments*, Nuclear Instruments and Methods in Physics Research Section A **1010**, 165461, (2020)

Additional material

Quantum Kernel Estimation:

- Maps classical data to a quantum Hilbert space.
- Quantum kernel measures similarity in high-dimensional space.

1. Quantum Support Vector Machines (QSVM)

Quantum Kernel Estimation:

- Maps classical data to a quantum Hilbert space.
- Quantum kernel measures similarity in high-dimensional space.

Quantum Kernel:

$$K(x, x') = |\left\langle \psi(x)|\psi(x')\middle|\psi(x)|\psi(x')\right\rangle|^2$$

Advantage: - Potentially exponential speedup over classical SVMs.

2. Quantum Neural Networks (QNNs)

Quantum Neural Networks replace classical neurons with parameterized quantum circuits.

Key Concepts:

- Quantum Gates as Activation Functions.
- Variational Quantum Circuits (VQCs) for optimization.

Quantum Neural Networks replace classical neurons with parameterized quantum circuits.

Key Concepts:

- Quantum Gates as Activation Functions.
- Variational Quantum Circuits (VQCs) for optimization.

Parameterized Quantum Circuit:

$$U(\theta) = \prod_i R_y(\theta_i) \cdot CNOT \cdot R_x(\theta_i)$$

Advantage: - Quantum gradients enable exploration of non-convex landscapes.

3. Quantum Boltzmann Machines (QBMs)

Quantum Boltzmann Machines leverage quantum mechanics to sample from a probability distribution.

- Quantum tunneling aids in escaping local minima.
- Quantum annealing for optimization problems.

3. Quantum Boltzmann Machines (QBMs)

Quantum Boltzmann Machines leverage quantum mechanics to sample from a probability distribution.

- Quantum tunneling aids in escaping local minima.
- Quantum annealing for optimization problems.

Quantum Hamiltonian:

$$H = -\sum_i b_i \sigma_i^z - \sum_{ij} w_{ij} \sigma_i^z \sigma_j^z$$

Advantage: - Efficient sampling in complex probability distributions.

Future Perspectives in QML

1. **Fault-Tolerant Quantum Computing:**
   - Overcoming noise for stable quantum circuits.
2. **Hybrid Quantum-Classical Models:**
   - Combining quantum circuits with classical neural networks.
3. **Quantum Internet:**
   - Distributed quantum machine learning over quantum networks.

Universal approximation theorem

The universal approximation theorem plays a central role in deep learning. Cybenko (1989) showed the following:

Let σ be any continuous sigmoidal function such that

$$\sigma(z) = \begin{cases} 1 & z \to \infty \\ 0 & z \to -\infty \end{cases}$$

Given a continuous and deterministic function $F(\boldsymbol{x})$ on the unit cube in d-dimensions $F \in [0,1]^d$, $x \in [0,1]^d$ and a parameter $\epsilon > 0$, there is a one-layer (hidden) neural network $f(\boldsymbol{x}; \boldsymbol{\Theta})$ with $\boldsymbol{\Theta} = (\boldsymbol{W}, \boldsymbol{b})$ and $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$, for which

$$|F(\boldsymbol{x}) - f(\boldsymbol{x}; \boldsymbol{\Theta})| < \epsilon \; \forall \boldsymbol{x} \in [0,1]^d.$$

The approximation theorem in words

Any continuous function $y = F(\boldsymbol{x})$ supported on the unit cube in d-dimensions can be approximated by a one-layer sigmoidal network to arbitrary accuracy.

Hornik (1991) extended the theorem by letting any non-constant, bounded activation function to be included using that the expectation value

$$\mathbb{E}[|F(\boldsymbol{x})|^2] = \int_{\boldsymbol{x}\in D} |F(\boldsymbol{x})|^2 p(\boldsymbol{x}) d\boldsymbol{x} < \infty.$$

Then we have

$$\mathbb{E}[|F(\boldsymbol{x}) - f(\boldsymbol{x};\boldsymbol{\Theta})|^2] = \int_{\boldsymbol{x}\in D} |F(\boldsymbol{x}) - f(\boldsymbol{x};\boldsymbol{\Theta})|^2 p(\boldsymbol{x}) d\boldsymbol{x} < \epsilon.$$

More on the general approximation theorem

None of the proofs give any insight into the relation between the number of of hidden layers and nodes and the approximation error ϵ, nor the magnitudes of $\boldsymbol{W}$ and $\boldsymbol{b}$.
Neural networks (NNs) have what we may call a kind of universality no matter what function we want to compute.

It does not mean that an NN can be used to exactly compute any function. Rather, we get an approximation that is as good as we want.

Class of functions we can approximate

The class of functions that can be approximated are the continuous ones. If the function $F(\boldsymbol{x})$ is discontinuous, it won't in general be possible to approximate it. However, an NN may still give an approximation even if we fail in some points.