Secrets revealed in this session:

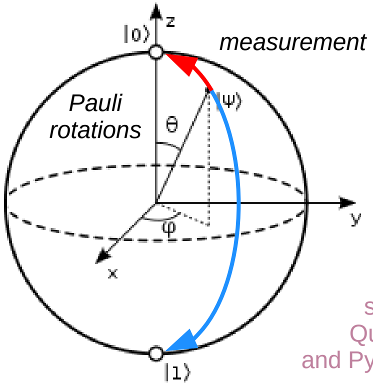
To explore the fundamental principles of parameterisation and optimisation of quantum models

Introduction to Quantum Machine Learning

Managing high complexity with the volume of data

Jacob L. Cybulski

Enquanted, Melbourne, Australia

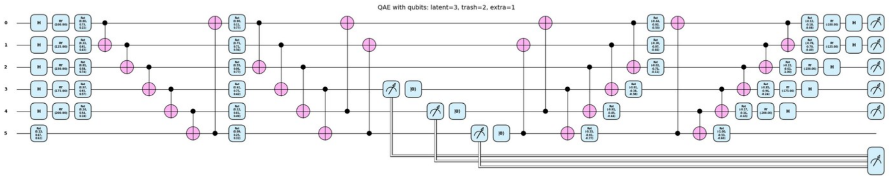


We will assume some knowledge of Quantum Computing and Python programming

11 October 2024, Deakin University, School of IT

Presenter

Jacob Cybulski
quantum@jacobcybulski.com



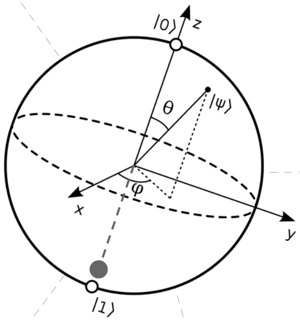
Founder
Researcher
Consultant
Author
at Enquanted

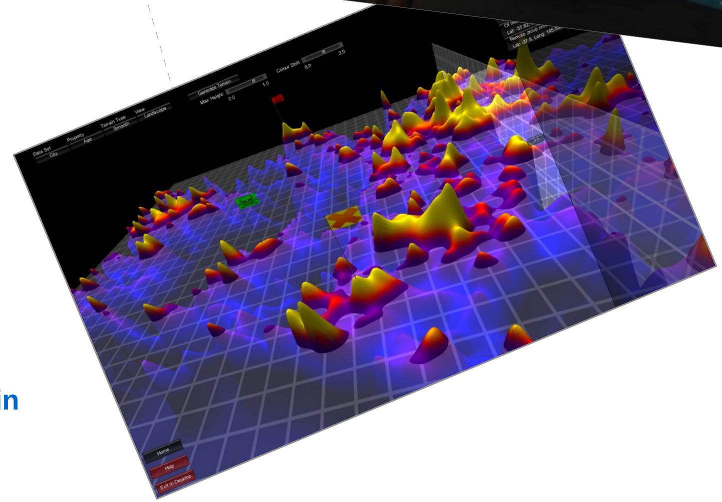
Research

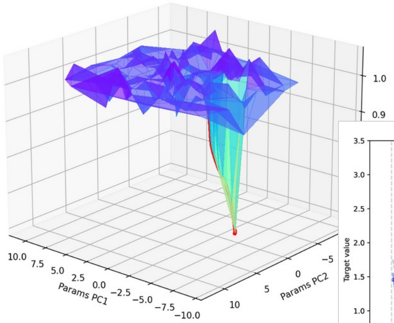
- Quantum computing
- Quantum machine learning
- Quantum time series analysis and anomaly detection
- Classical machine learning
- Data visualisation

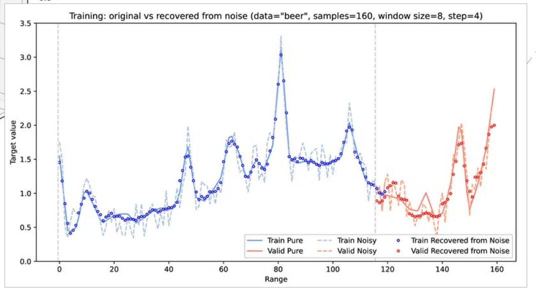
Personal

- Recreational cycling
- Reading science and Sci-Fi
- Quantum challenges and hackathons



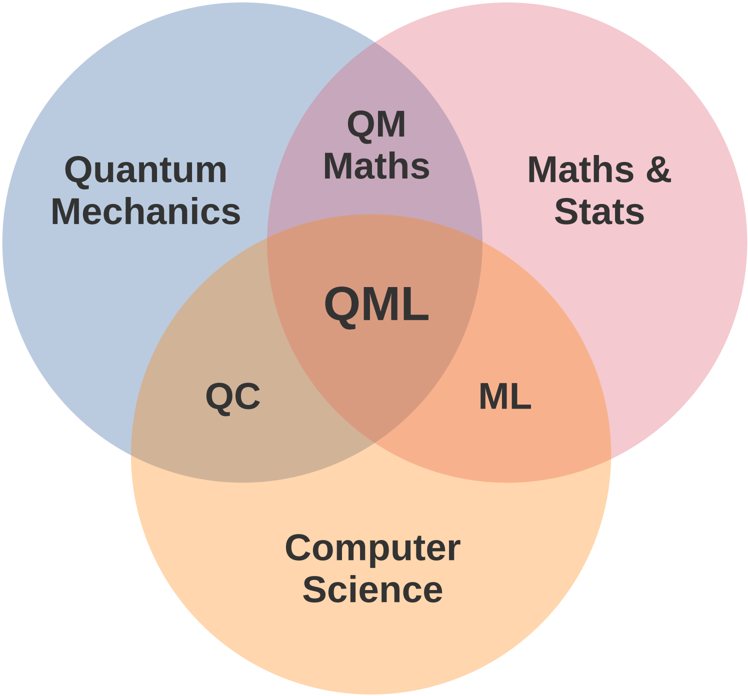






Research collaboration and supervision of research students in QC + QML

What is QML?



Quantum Machine Learning:

a discipline seeking to take advantage of quantum mechanical processes to induce or enhance machine learning

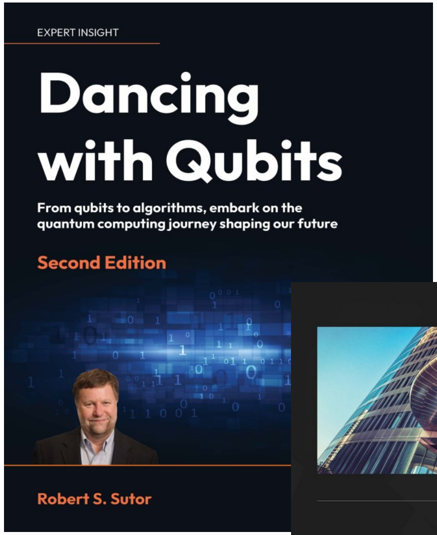
QML combines in novel ways the concepts and methods adopted from:

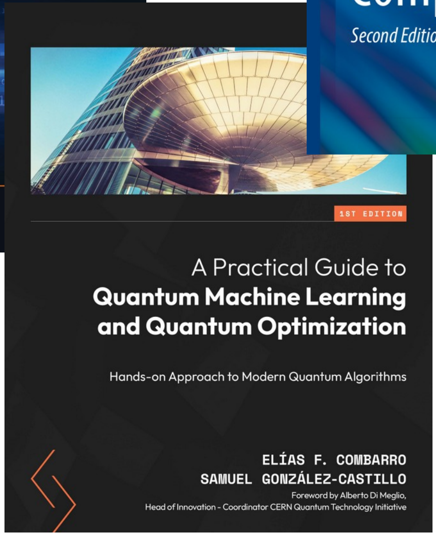
- Quantum Computing (QC)
- Machine Learning (QC)
- Quantum Mechanics (QM)

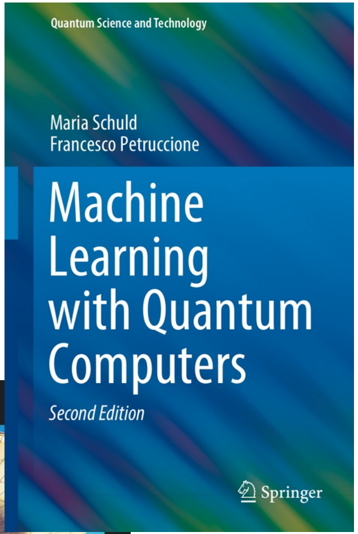
QML exploits unique properties and behaviour of quantum systems to improve computation, due to:

- Superposition and entanglement
- Exponential size of the quantum state space
- Linearity of quantum models
- Reversibility of unmeasured quantum models

Recommended reading
on QML








PennyLane: Automatic differentiation of hybrid quantum-classical computations

Ville Bergholm,[1] Josh Izaac,[1] Maria Schuld,[1] Christian Gogolin,[1] M. Sohaib Alam,[2] Shahnawaz Ahmed,[3] Juan Miguel Arrazola,[1] Carsten Blank,[4] Alain Delgado,[1] Soran Jahangiri,[1] Keri McKiernan,[2] Johannes Jakob Meyer,[5] Zeyue Niu,[1] Antal Száva,[1] and Nathan Killoran[1]

[1]Xanadu, 777 Bay Street, Toronto, Canada
[2]Rigetti Computing, 2919 Seventh Street, Berkeley, CA 94710
[3]Wallenberg Centre for Quantum Technology, Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden
[4]data cybernetics, Martin-Kolmsperger-Str 26, 86899 Landsberg, Germany
[5]Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany

14 Feb 2020


Modern applications of machine learning in quantum sciences

Anna Dawid[1,2*], Julian Arnold[3†], Borja Requena[2†], Alexander Gresch[4†], Marcin Płodzień[2], Kaelan Donatella[5], Kim A. Nicoli[6,7], Paolo Stornati[2], Rouven Koch[8], Miriam Büttner[9], Robert Okuła[10,11], Gorka Muñoz–Gil[12], Rodrigo A. Vargas–Hernández[13,14], Alba Cervera-Lierta[15], Juan Carrasquilla[14], Vedran Dunjko[16], Marylou Gabrié[17], Patrick Huembeli[18,19], Evert van Nieuwenburg[16,20], Filippo Vicentini[18], Lei Wang[21,22], Sebastian J. Wetzel[23], Giuseppe Carleo[18], Eliška Greplová[24], Roman Krems[25], Florian Marquardt[26,27], Michał Tomza[1], Maciej Lewenstein[2,28] and Alexandre Dauphin[2*]

1 Faculty of Physics, University of Warsaw, Poland
2 ICFO - Institut de Ciències Fotòniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain
3 Department of Physics, University of Basel, Switzerland
4 Quantum Technology Research Group, Heinrich-Heine-Universität Düsseldorf, Germany
5 Université de Paris, CNRS, Laboratoire Matériaux et Phénomènes Quantiques, France
6 Machine Learning Group, Technische Universität Berlin, Germany
7 BIFOLD, Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germany
8 Department of Applied Physics, Aalto University, Espoo, Finland
9 Institute of Physics, Albert-Ludwig University of Freiburg, Germany
10 International Centre for Theory of Quantum Technologies, University of Gdańsk, Poland
11 Department of Algorithms and System Modeling, Faculty of Electronics, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Poland
12 Institute for Theoretical Physics, University of Innsbruck, Austria
13 Department of Chemistry, University of Toronto, Canada
14 Vector Institute for Artificial Intelligence, MaRS Centre, Toronto, Canada
15 Barcelona Supercomputing Center, Spain
16 LIACS, Leiden University, The Netherlands
17 CMAP, École Polytechnique, France
18 Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
19 Menten AI, Inc., Palo Alto, California, United States of America
20 Niels Bohr Institute, Copenhagen, Denmark
21 Beijing National Lab for Condensed Matter Physics and Institute of Physics, Chinese Academy of Sciences, Beijing, China
22 Songshan Lake Materials Laboratory, Dongguan, China
23 Perimeter Institute for Theoretical Physics, Waterloo, Canada
24 Kavli Institute of Nanoscience, Delft University of Technology, NL-2600 GA Delft, The Netherlands
25 Department of Chemistry, University of British Columbia, Vancouver, Canada
26 Max Planck Institute for the Science of Light, Erlangen, Germany
27 Department of Physics, Friedrich-Alexander Universität Erlangen-Nürnberg, Germany
28 ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain
† These authors contributed equally.
* Anna.Dawid@fuw.edu.pl, Alexandre.Dauphin@icfo.eu

June 23, 2022

Abstract

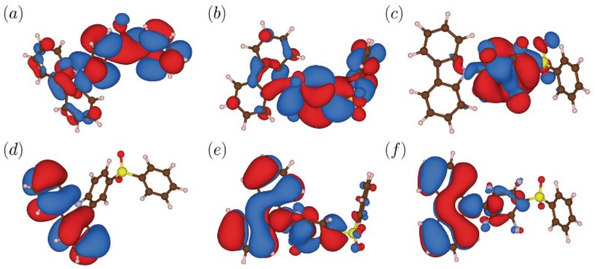
In these Lecture Notes, we provide a comprehensive introduction to the most recent advances in the application of machine learning methods in quantum sciences. We cover the use of deep learning and kernel methods in supervised, unsupervised, and reinforcement learning algorithms for phase classification, representation of many-body quantum states, quantum feedback control, and quantum circuits optimization. Moreover, we introduce and discuss more specialized topics such as differentiable programming, generative models, statistical approach to machine learning, and quantum machine learning.

1

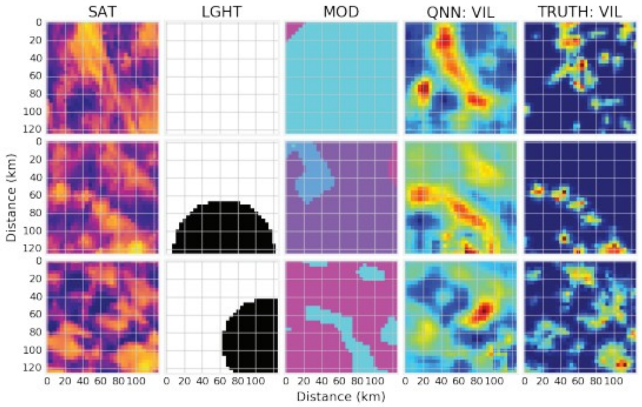
For more recent examples, see:
Olivier Ezratty, 2024. Understanding Quantum Technologies, 7th ed.
Le Lab Quantique. URL: https://www.oezratty.net/

Three Selected Applications

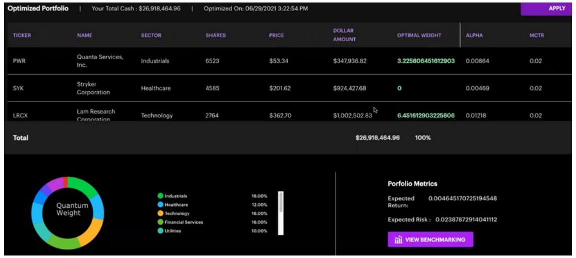
- **Chemistry**
- **Company:** Mitsubishi, IBM, and partners, 2021
- **Platform:** IBM quantum machine with Qiskit and Quantum Chemistry toolkit
- **Aim:** increase efficiency of Organic Light Emitting Diodes (OLED) to 100% (now only 25%).
- **Results:** Predicted exact properties of OLED materials to improve efficiency.

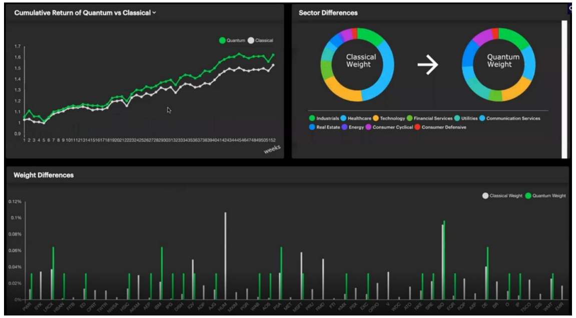


- **Weather Radar**
- **Company:** Rigetti, 2021
- **Platform:** Rigetti quantum machine with Quil, using a QNN.
- **Aims:** synthetic weather radar images, produced without radar.
- **Results:** A hybrid classical-quantum storm prediction – an improvement over classical machine learning.

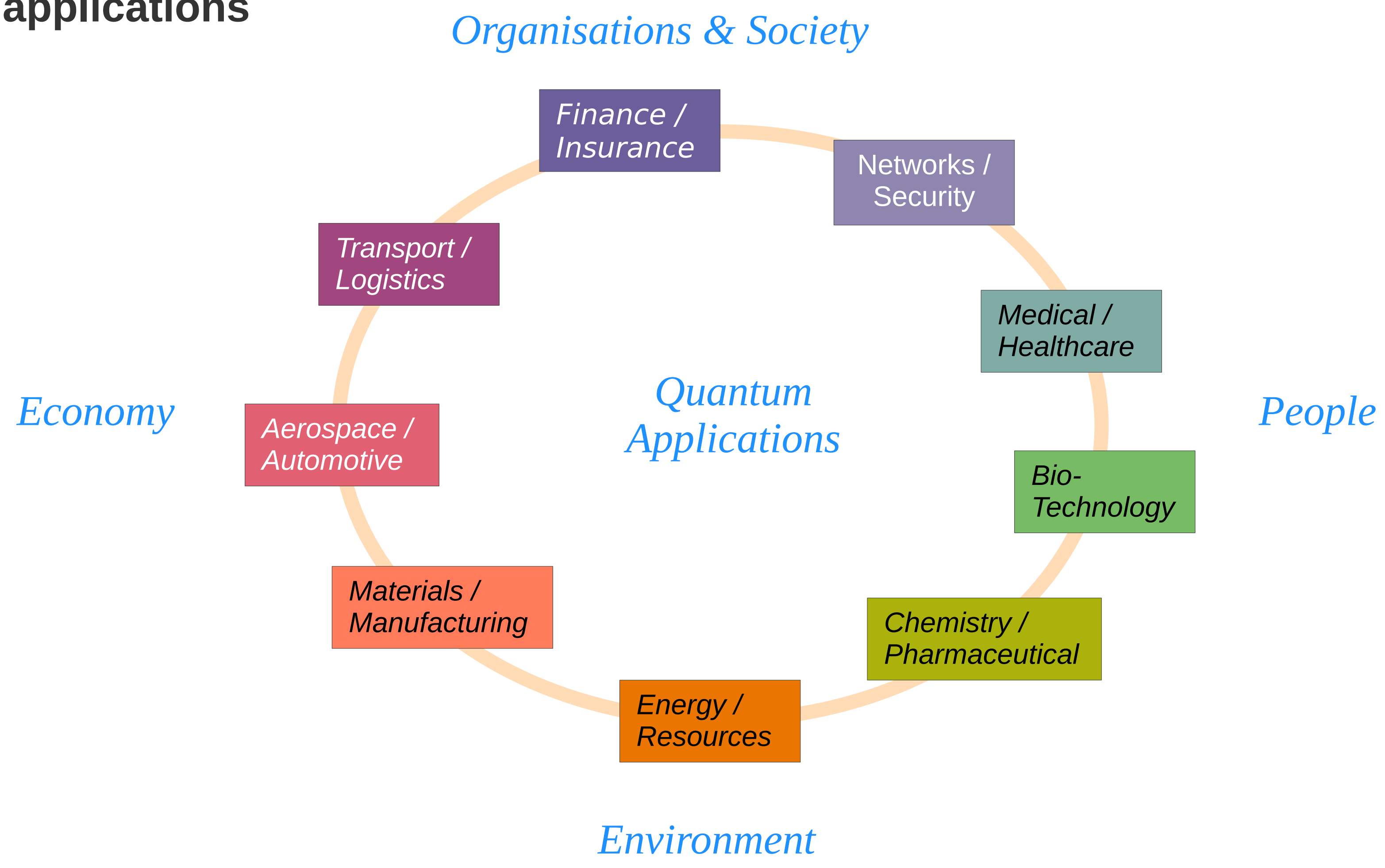


- **Finance**
- **Company:** Accenture, 2021
- **Platform:** D-Wave quantum machine with Leap, via AWS Braket
- **Aim:** to minimise the difference between the target and the final portfolio while maximising the return, using data from Yahoo Finance.
- **Results:** Working portfolio rebalancing system.





The wheel of
QML applications



QML platforms / SDKs

Qiskit, PennyLane, Cirq, Yao, ...

- **Other Platforms / Q-SDKs**
  Classiq / Classiq, Forest / Rigetti, Ocean / D-Wave, Quantum Development Kit with Q# / Microsoft, cuQuantum / Nvidia, t|ket> / CQC

- **Qiskit** (OS)
  - **Location:** USA
  - **Language:** Python
  - **Company:** IBM Research
  - **Backends:** IBM, AQT, IQM, Rigetti, Quantinuum
  - **Models:** VQC, VQR, QNN, QCNN, QSVM, QGAN, Q Kernels, VQE, VQLS, QFT, QAOA
  - **ML SDK:** Scipy, PyTorch, Tensorflow
  - **Apps:** QML, Finance, Optimization, Nature

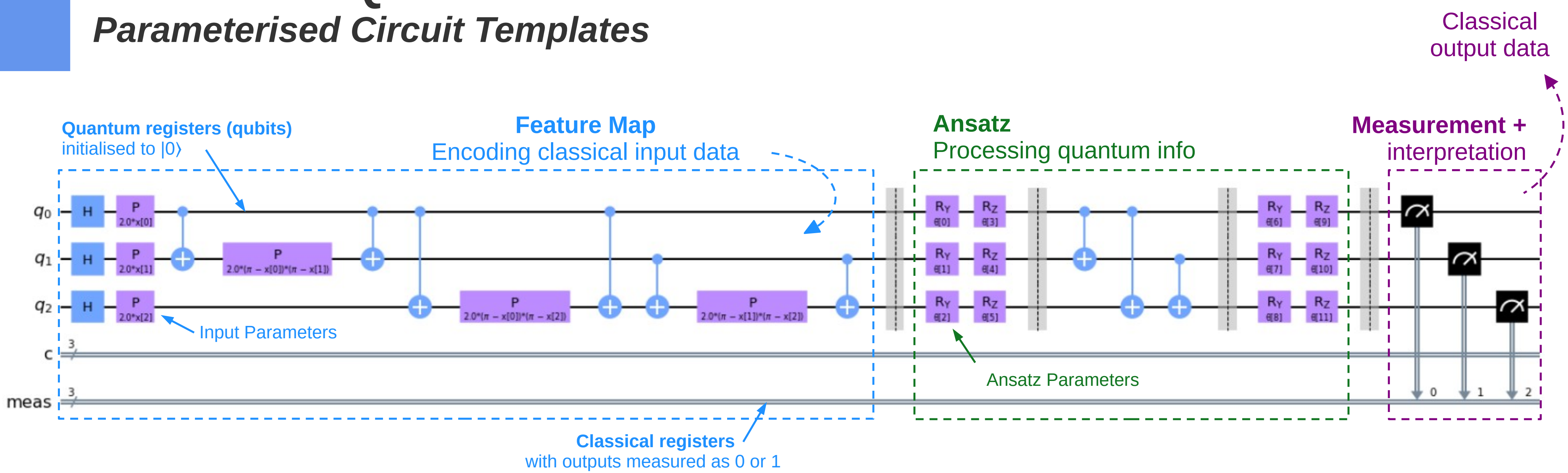
- **PennyLane** (OS)
  - **Location:** Canada
  - **Language:** Python
  - **Company:** Xanadu
  - **Backends:** Xanadu, AQT, IonQ, Rigetti, Honeywell
  - **Models:** QNN, Q Kernels, QFT, QAOA ...
  - **ML SDK:** PyTorch, Tensorflow
  - **Apps:** QML. Optimization, Chemistry

- **Cirq** (OS)
  - **Location:** USA
  - **Language:** Python
  - **Company:** Google Quantum AI
  - **Backends:** Google, AQT, IonQ, Pasqal, Rigetti
  - **Models:** VQE, QAOA, via TF Quantum (QNN, QCNN, QRNN, QGNN, QGAN, QRL, Q kernels)
  - **ML SDK:** PyTorch, Tensorflow
  - **Apps:** QML, Chem, Materials, Comms, Metrology

- **Yao** (OS)
  - **Location:** China / Taiwan
  - **Language:** Julia
  - **Company:** QuantumBFS
  - **Backends:** Simulators, via Python
  - **Models:** VQE, many others via Julia (Flux)
  - **ML SDK:** via Julia/Python (scipy, sklearn, Tensorflow)
  - **Apps:** Via Julia (QML, AI, Optimization, Physics, Chemistry, Biology, Earth, Finance, Robotics)

Variational Quantum Models =

Parameterised Circuit Templates



Quantum circuits are static

Data and operations are hard-coded

New data / operation params → new circuit

Typically, the circuit consists of three blocks:

- a feature map (input)
  encodes classical data as circuit state
- an ansatz (processing)
  alters circuit state
- measurements (output)
  measures circuit state into classical data

Variational Quantum Algorithm

VQA is an ***iterative process***

VQA uses ***cost/loss function*** and ***optimiser***

VQA has ***difficulties***:

- The problem at hand
- Large circuits with many parameters
- Complex measurement strategy
- Unsupervised learning
- Emergence of barren plateaus

A typical VQA process

The ansatz parameters are initialised to some values, e.g. zero or random

The feature map parameters are bound to the new input data

The parameter values are used to create a new circuit

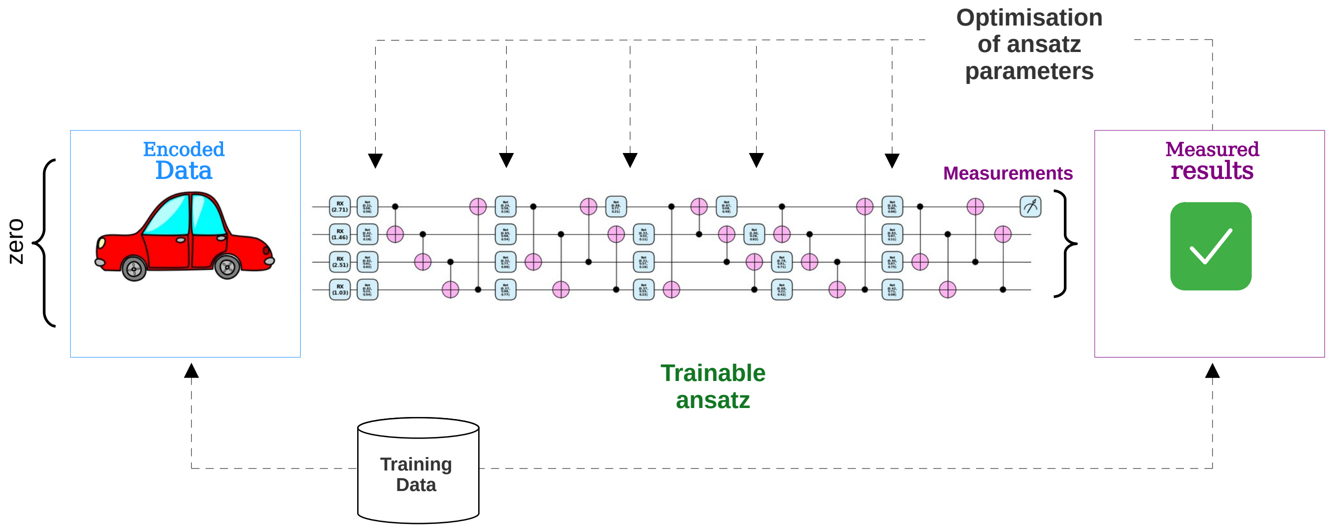
The circuit is executed

The circuit quantum state is then measured

Cost function is applied to measurement results and expected values

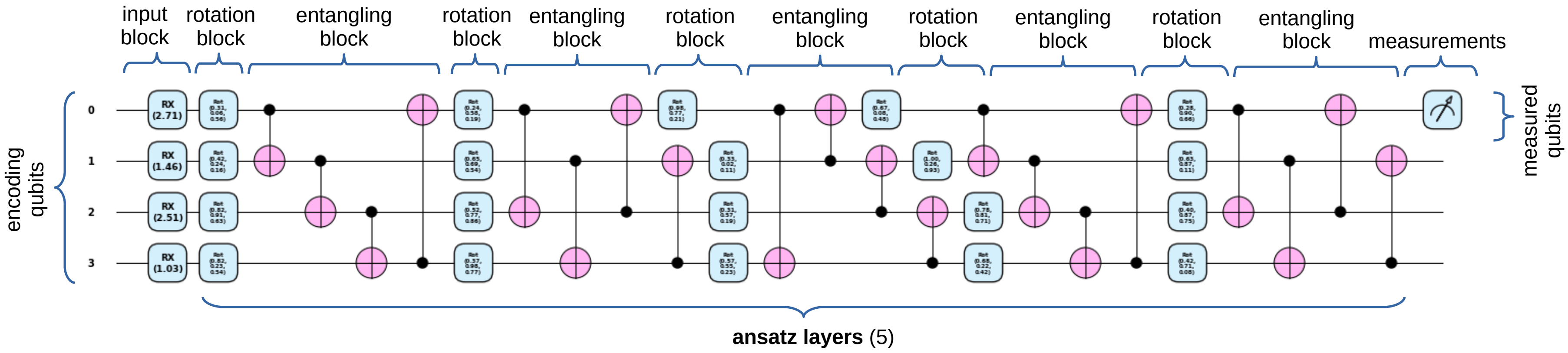
The cost of difference is calculated

Based on the difference and previous parameters the new parameters values are proposed



Ansatz design and optimisation

A simple quantum classifier to start with...



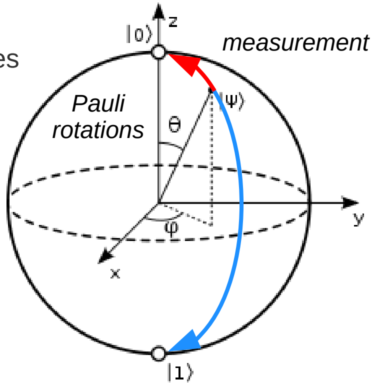
feature maps vary in:
structure and function

ansatze vary in:

- width (qubits #)
- depth (layers #)
- dimensions (param #)
- structure (e.g. funnelling)
- entangling (circular, linear, sca)

ansatz layers consist of:
rotation blocks and entangling blocks of R(x, y, z) and CNOT gates

rotation gates
alter qubit states around x, y, z axes



different cost functions:
R2, MAE, MSE, Huber, Poisson, cross-entropy, hinge-embedding, Kullback-Leibner divergence

different optimisers:
gradient based (Adam, NAdam and SPSA)
linear approximation methods (COBYLA)
non-linear approximation methods (BFGS)

cost and optimisation depend on:
task, ansatz design, measurement strategy, training data and platform

From the lab to the world?

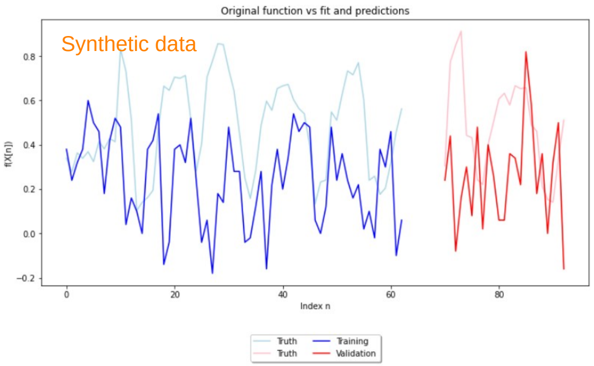
a story of one quantum TS analysis model...

An area of Jacob's research

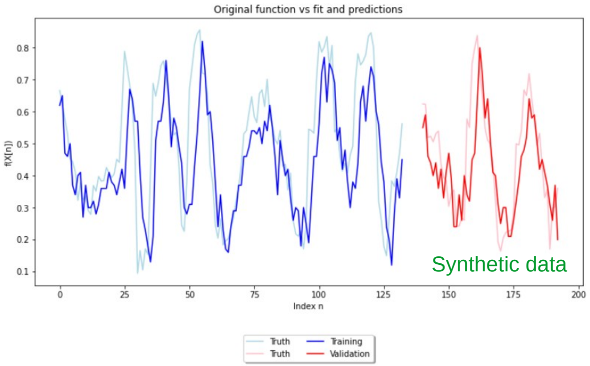
Sliding Window QNN

Synthetic data

→ worked well only for simple data sets



over 100 experiments



Sliding Window Serial Model

Synthetic data

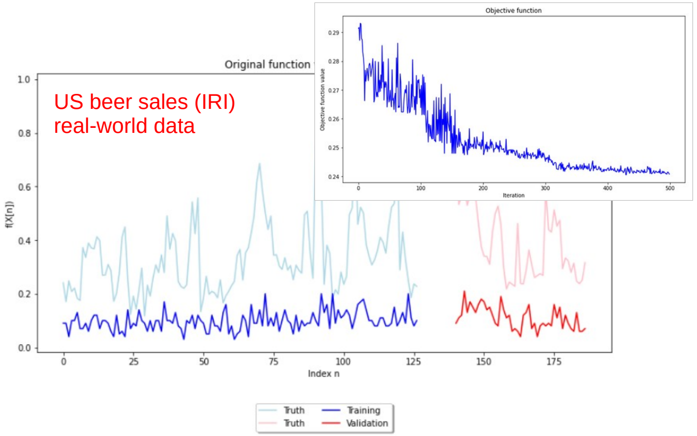
→ good prediction of both seasonality and the signal amplitude.

Ready for the real-world data?

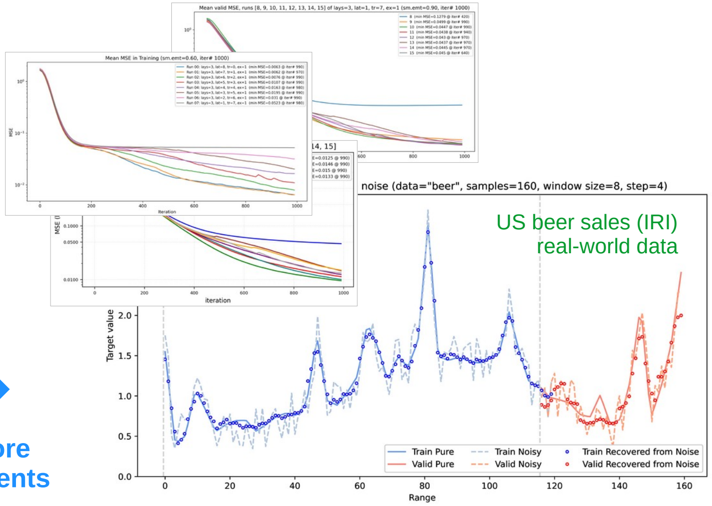
Sliding Window Serial Model

Real-world data

beer sales in USA → the model failed completely!



500 more experiments



Sliding Window Serial Model

Real-world data

500 experiments with varying parameters + changes to the model + cost fun and optimiser + changed the platform to PennyLane →

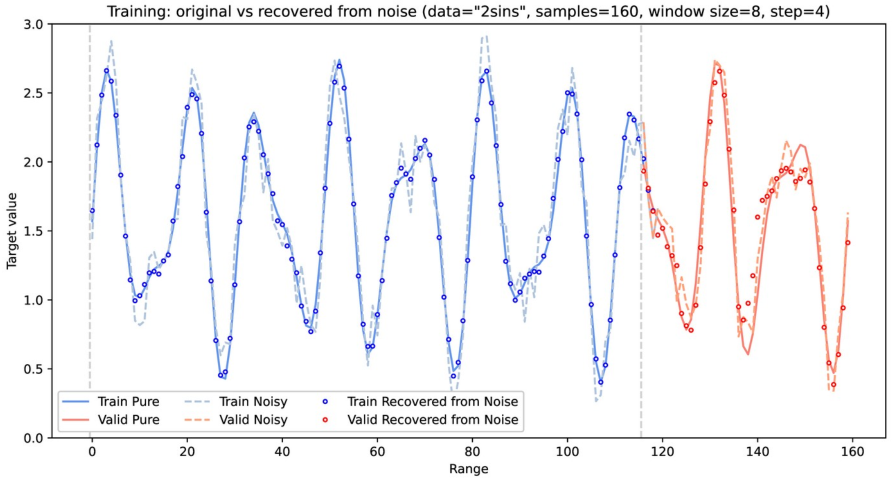
good prediction + noise elimination

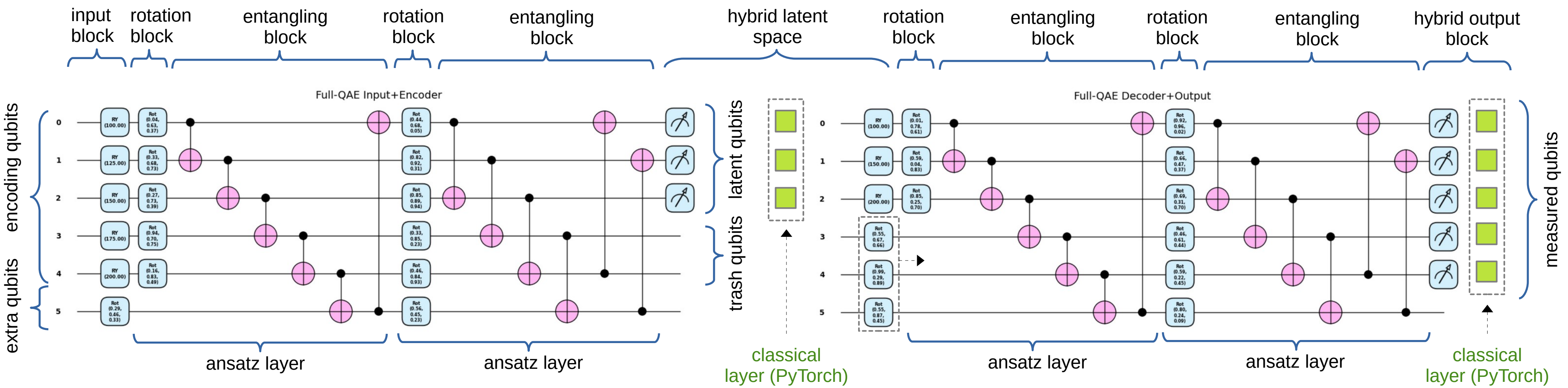
QAE for Time Series

An area of Jacob's research

One of Jacob's projects is development of complex quantum models, quantum and hybrid, for time series and signals. The models capable of cleaning data, analysing and forecasting temporal data and anomaly detection.

Sample applications include: machine condition monitoring, astronomical observations, nationwide marketing and sales, earthquake prediction, EEG or ECG analysis, etc.





Olivier Ezratty, Understanding Quantum Technologies, 7th Ed (2024)

Maria Schuld and Francesco Petruccione. *Machine Learning with Quantum Computers. 2nd ed*. Springer, 2021.

QML

advice on model development

Experiments:
perform experiments with different platforms, architectures, cost functions, optimisers, with multiple approaches to model initialisations

Data encoding and decoding:
ensure they are consistent, also in terms of the adopted approach to measurements and their interpretation

design methods of measuring and interpreting model's quantum state,as they are essential for model training and testing

Ansatz architecture:
plan the ansatz circuit width, depth, the number of layers and trainable parameters, extra degrees of freedom (extra params), as they all determine the model performance

Statistics:
collect stats, average performance and deviations, plot results to compare models, also against equivalent classical solutions, in model training, validation and testing

Process:
just because you are using quantum computing methods, it does not mean you can skip the traditional data science diligence and good software development practices - just the opposite!

Stop developing
Start reusing!

Try existing QML models and algorithms:

- Quantum Neural Networks (QNN, VQC/R, QCNN, qGAN)
- Quantum Kernel Methods (Feature Maps, Estimators)
- Quantum Optimisation Algorithms (QAOA, QUBO)
- Quantum Support Vector Machines (QSVM, QSVC/R)
- Quantum Clustering Algorithms (QCA k-NN, DQC)
- Quantum Fourier Analysis (QFT, QFFT)
- Quantum Sequence Models (QRNN, QLSTM, QGRU)
- Quantum Annealing / Quantum Adiabatic Algorithm (QAA)
- Quantum Boltzmann Machines (QBM, QRBM))
- Quantum Principal Components Analysis (QPCA)
- Quantum Self-Attention and Transformers
- Quantum Random Forest (QRF)
- Quantum k-Nearest Neighbour (QkNN)
- Quantum Hopfield Associative Memory (QHAM)
- Quantum Reinforcement Learning (QRL)
- Quantum Bayesian Modelling (QBN, QBC, QBNN)
- Quantum Genetic Algorithms (QGA)

Alternative Technologies

Accelerators, Hybrids, Tensor Networks, Brain-Inspired Systems

PennyLane/ PyTorch
approach to model development

One of the issues found in Qiskit:

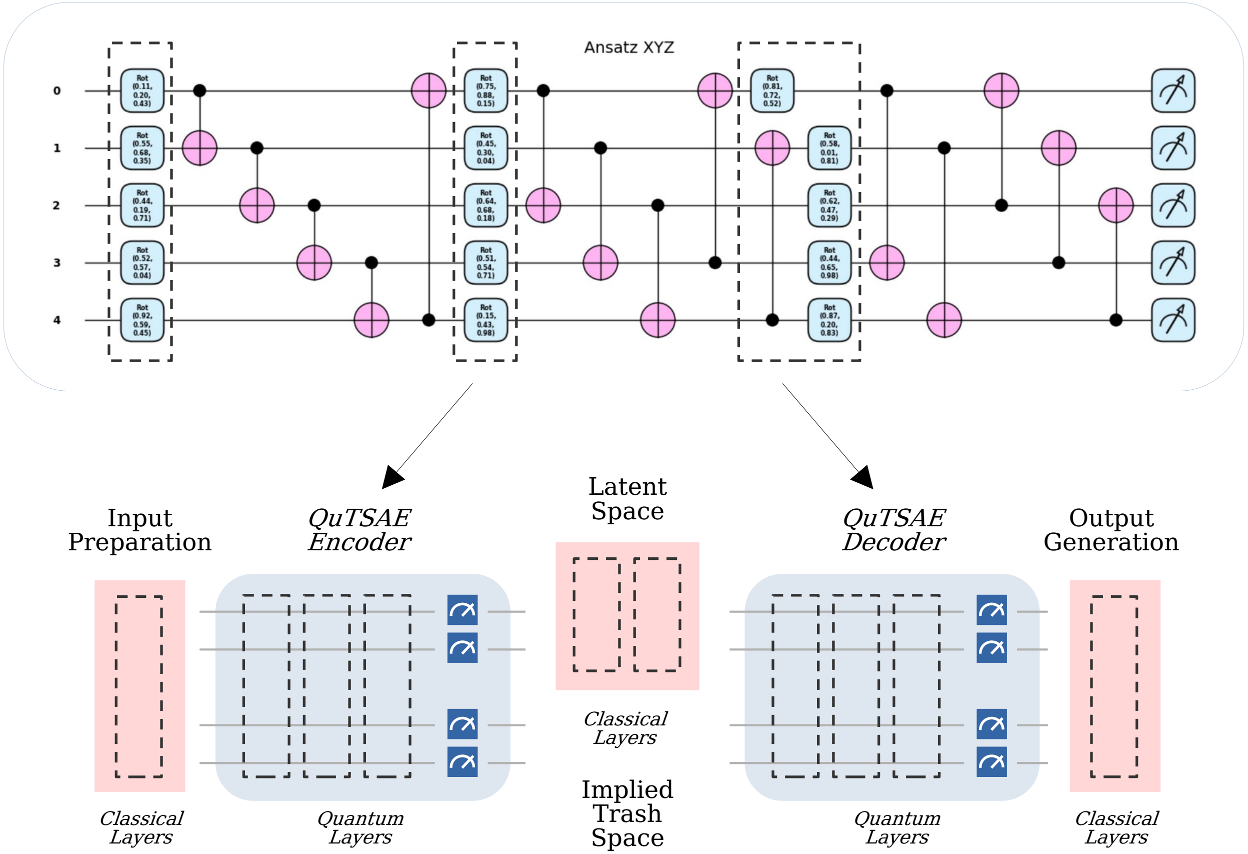
- the lack of quantum models' transparency which hampered the performance of gradient optimisers

Solution - PennyLane / PyTorch, with:

- hybrid models of quantum / classical components
- parameters structured into layers, trained very efficiently with gradient optimisers

Technologies alternative to quantum:

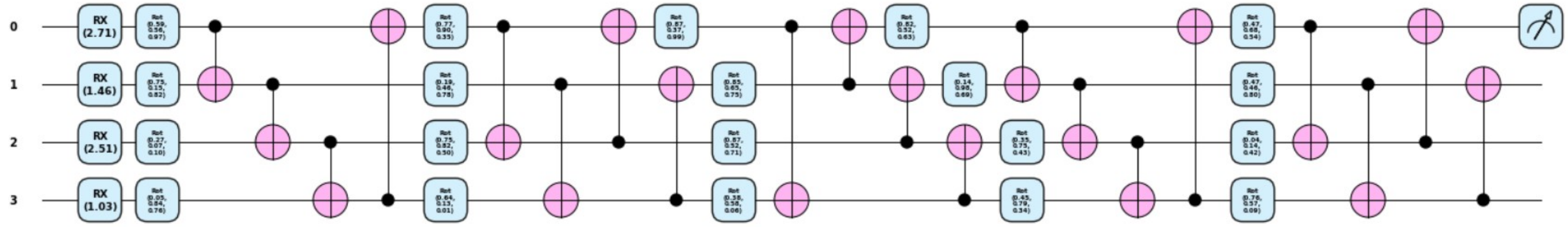
- GPU and TPU accelerators
- hybrid quantum-classical solutions
- quantum inspired models
- tensor networks
- neuromorphic and neuro-inspired systems



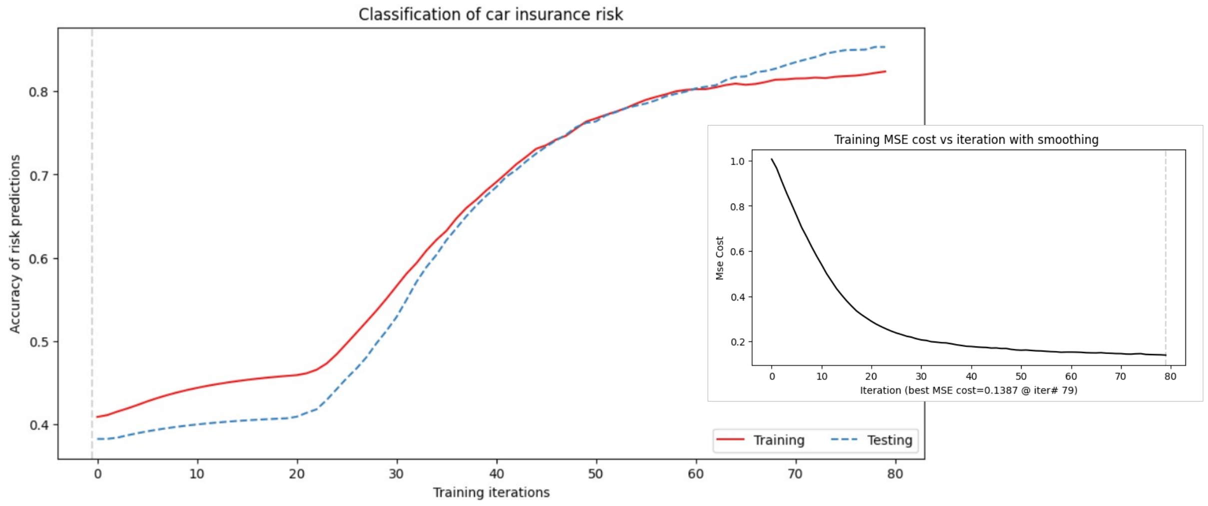
PennyLane / PyTorch Neural Network
of classical and quantum components

Demo:

Automotive insurance risk assessment (quantum classification)




```
 0 (000007 sec): Loss      1   Acc 0.4015
 4 (000034 sec): Loss 0.7645   Acc 0.4526
 8 (000062 sec): Loss 0.5765   Acc 0.4672
12 (000089 sec): Loss 0.4288   Acc 0.4672
16 (000116 sec): Loss  0.338   Acc 0.4672
20 (000144 sec): Loss 0.2746   Acc 0.4745
24 (000171 sec): Loss  0.233   Acc 0.5693
28 (000198 sec): Loss 0.2126   Acc 0.6715
32 (000226 sec): Loss 0.1947   Acc 0.7226
36 (000253 sec): Loss  0.182   Acc 0.7737
40 (000280 sec): Loss 0.1784   Acc 0.7883
44 (000307 sec): Loss 0.1704   Acc 0.8102
48 (000335 sec): Loss 0.1631   Acc 0.8175
52 (000362 sec): Loss 0.1621   Acc 0.8175
56 (000389 sec): Loss 0.1589   Acc 0.8102
60 (000417 sec): Loss 0.1522   Acc 0.8321
64 (000444 sec): Loss  0.149   Acc 0.8467
68 (000471 sec): Loss 0.1473   Acc 0.8248
72 (000499 sec): Loss 0.1428   Acc 0.8321
76 (000526 sec): Loss 0.1449   Acc 0.8102
```



Thank you!

Any questions?

Photos from Unsplash
Enquanted is being somewhere in-between Enchanted and Entangled