This session:

Observer's gaze shifts,
Collapsing time's vast ocean—
A single ripple.

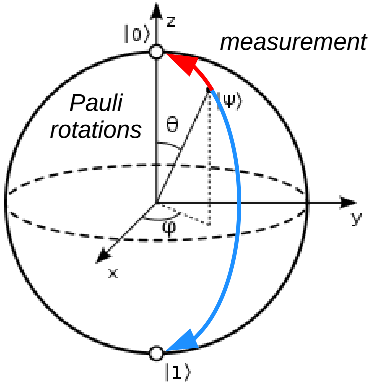
TQM Haiku by DeepSeek

Temporal Quantum Models (TQM)
  - *measuring and interpreting*
The problem of quantum time
Apps, devs and issues (with QML)
Measurements and interpretations
  - *global and local cost,*
  - *amps, probs and expectations*
Training temporal quantum models
The curse of dimensionality
  - *barren plateaus and sparse outcomes*
Rethinking temporal quantum models
  - *frugal ansatz design, BP resistance,*
  - *classical and analog alternatives*
Summary, current work and questions

Why Temporal Quantum Models (TQM) are so damn hard to train !?!

Mainly practical issues with some theoretical overtones

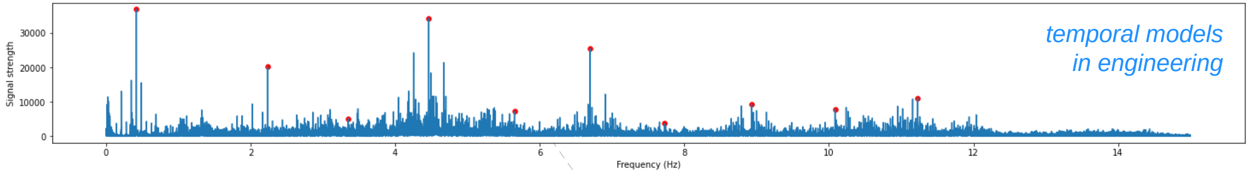
Jacob L. Cybulski
Enquanted, Melbourne, Australia
Hon. Assoc. Prof. Deakin University, Australia



15 March 2025, 5th Quantum Machine Learning Conference

Dashboard

Jacob Cybulski
quantum@jacobcybulski.com



temporal models
in engineering

Founder
Researcher
Consultant
Author
at Enquanted

also

Honorary A/Prof
In Quantum Computing
School of IT, SEBE
Deakin University
Melbourne, Australia

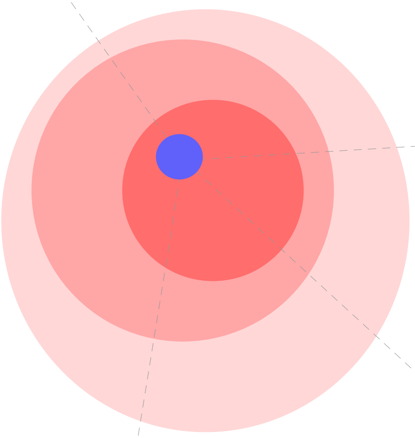
Research

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

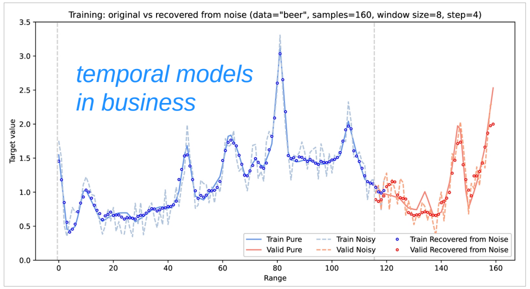
Personal

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

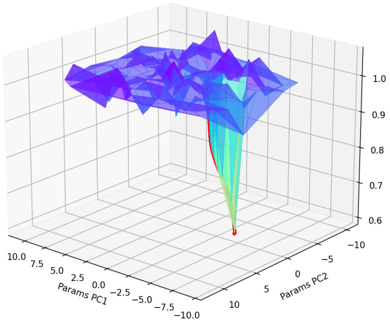
Also, supervision of
student projects,
open science projects,
QC/QML promotion, etc.



temporal models
in medicine



temporal models
in business



temporal models
training difficulties

The problem of time

Tempus rerum imperator

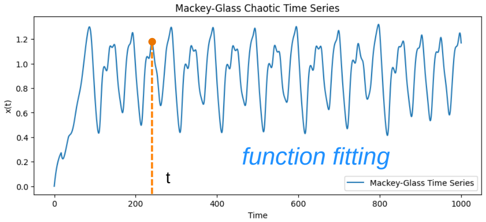
$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \hat{H}\Psi(\mathbf{r},t)$$

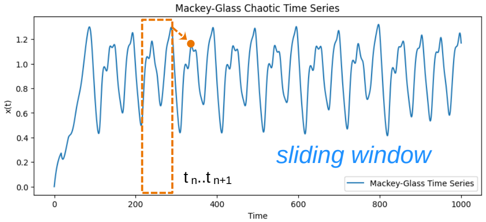
Quantum Computing (QC), similarly to **Quantum Mechanics (QM)**, takes a “pragmatic” approach to ***time***, which is considered an external variable presiding over ***changes*** in observable phenomena. Our clocks are ticking in a flat Universe measured in classical intervals of our watches.

QC models can therefore be developed to analyse ***time*** and ***change*** of the world phenomena.

Temporal quantum models (TQM) are QC models able to deal with time and change, represented in data, but encoded into the model structure, e.g.

- by using time as a ***parameter*** to guide the function fitting of time-ordered data (left)
- by casting a problem into a ***timeless form*** describing changes as relations in data representing the past and the future (right).





Apps, devs and issues

QML to the rescue

Sample TQM applications

Apps are found in Sci & Eng, Earth Sci, Finance, Meds, etc.

- Explanation (sequence to function)
- Decision support (sequence to logical)
- Forecasting and anomaly detection (sequence to points)
- Monte Carlo + Random walks (constraints to sequence)
- Noise and anomaly elimination (sequence to sequence)

Dev issues with temporal quantum modelling

In the majority of applications, temporal quantum models and their circuits potentially suffer from a number of problems, e.g.

- Temporal data has unique characteristics, e.g. tacit features, volume, continuity, cyclicity, noise, anomalies, volatility, etc.
- High complexity of non-trivial cases (features / anomalies)
- Large quantum models / circuits
- Low learning capacity
- High training difficulties
- High error rates on NISQ machines
- Competition from classical approaches
- Competition from quantum-inspired approaches

QML algorithms that deal with time and change

Time in data

- Quantum Sequence Models (QRNN, QLSTM, QGRU)
- Quantum Reservoir Computing (QRC)
- Quantum Self-Attention and Transformers (LLMs)
- Quantum Fourier Analysis (QFT, PQFT, QFFT)

Change and state evolution

- Quantum Optimisation Algorithms (QAOA, QUBO)
- Quantum Annealing / Quantum Adiabatic Algorithm (QAA)
- Quantum Reinforcement Learning (QRL)
- Quantum Bayesian Modelling (QBN, QBC, QBNN)
- Quantum Genetic Algorithms (QGA)

Supporting models

- Quantum Neural Networks (QNN, VQC/R, QCNN, qGAN)
- Quantum Support Vector Machines (QSVM, QSVC/R)
- Quantum Kernel Methods (Feature Maps, Estimators)
- Quantum Clustering Algorithms (QCA k-NN, DQC)

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

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

Commonly used

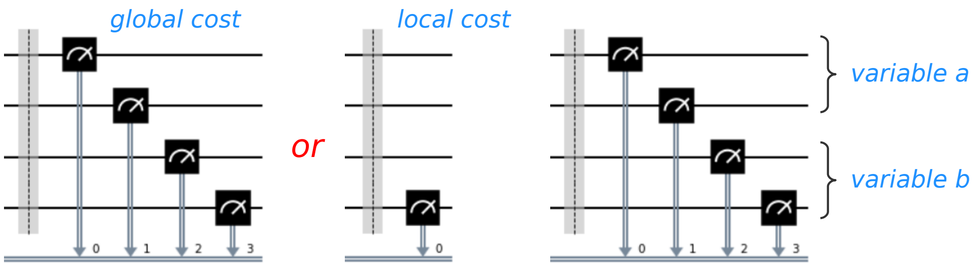
measurements and interpretation

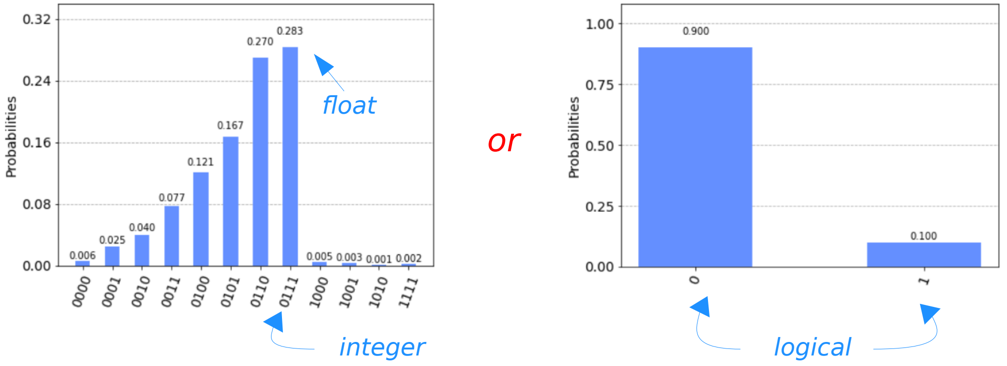
There are many ways of obtaining the outcome of a circuit execution, e.g. we can measure:

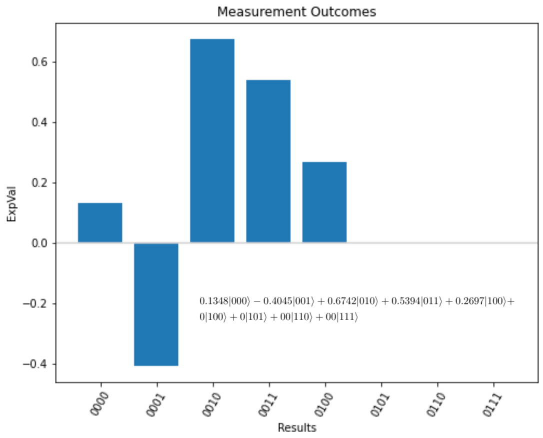
- all qubits (global cost / measurement)
- a few selected qubits (local cost / measurement)
- groups of qubits (each as a variable value)
- as counts of outcomes (repeated measurements)
- as probabilities of outcomes (e.g. P(|0111⟩))
- as Pauli expectation values (i.e. of eigenvalues)
- as expectation of interpreted values (e.g. 0 to 15)
- as variance, etc.

Repeated circuit measurement can be interpreted as outcomes of different types, e.g.

- as a probability distribution (as is)
- as a series of values (via expvals)
- as a binary outcome:
  single qubit measurement or parity of kets
- as an integer:
  most probable ket in multi-qubit measurement
- as a continuous variable:
  probability of the selected ket (e.g. |0ⁿ⟩)







Or we can measure expectation values of the circuit state and interpret them as a series of values

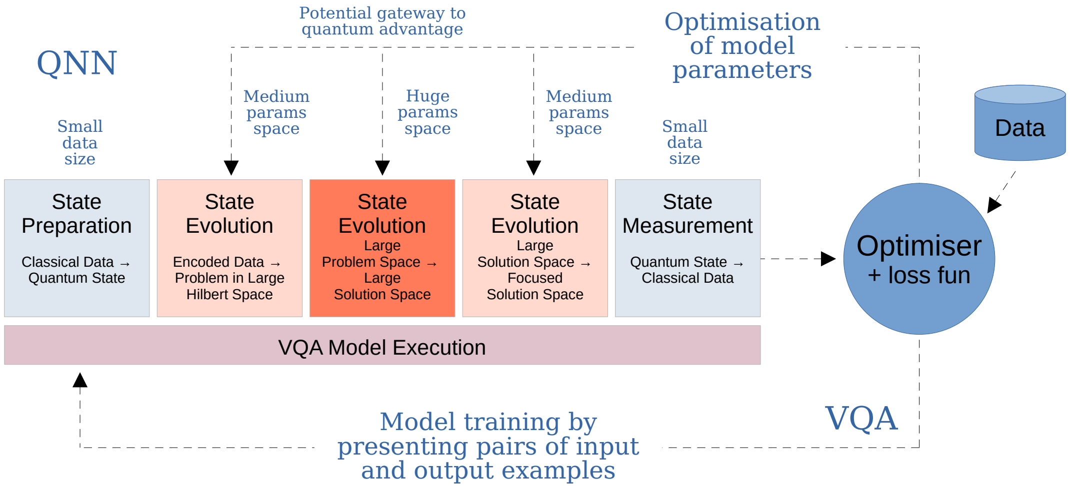
So we can represent and measure almost anything!

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

Training temporal quantum models

Their baits and traps

- TQMs are trained using Variational Quantum Algorithms (VQA).
- TQMs hence utilise quantum model execution and classical parameters optimisation using a classical loss/cost function.
- TQMs are designed to evolve a problem state into a solution state that can be sampled with measurements.



Potential gateway to quantum advantage

Optimisation of model parameters

QNN

Small data size

Medium params space

Huge params space

Medium params space

Small data size

Data

State Preparation
Classical Data → Quantum State

State Evolution
Encoded Data → Problem in Large Hilbert Space

State Evolution
Large Problem Space → Large Solution Space

State Evolution
Large Solution Space → Focused Solution Space

State Measurement
Quantum State → Classical Data

Optimiser + loss fun

VQA Model Execution

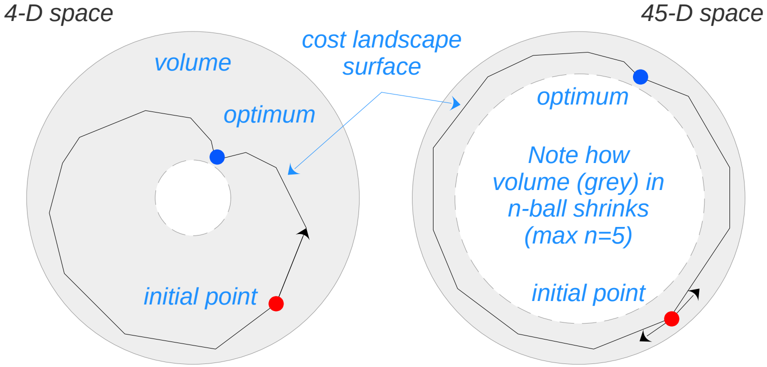
Model training by presenting pairs of input and output examples

VQA

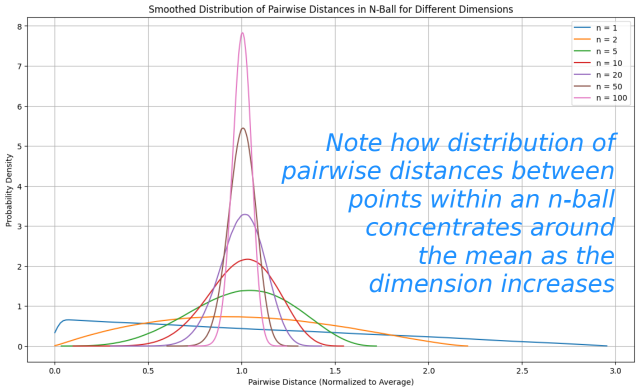
- TQMs are often designed as circuits of many qubits, layers and trainable parameters, able to process complex data, but consequently creating training difficulties, e.g.
  - ***High-dimensional parameter space*** (large circuits)
    ☑ enables linear separation of quantum information, but
    ☒ flattens the gradient space.
  - ***Highly entangled circuits*** (complex circuits)
    ☑ inter-relate qubits and their training parameters, but
    ☒ suffer from non-local gradients, complex state dynamics, cost landscape complexity, and high decoherence rate.
  - ***Global cost*** (measuring all qubits)
    ☑ allows better utilisation of model training parameters, but
    ☒ requires exponential increase of the circuit runs, needed to prevent sparse distribution of outcomes.
  - ***Model initialisation*** (optimisation starting point)
    ☑ quick, easy and popular random params initialisation, but
    ☒ proven to make model optimisation ineffective.
  - *All of the above are said to attract barren plateaus*!

So what are those scary barren plateaus?

The curse of dimensionality



Note how volume (grey) in n-ball shrinks (max n=5)



Note how distribution of pairwise distances between points within an n-ball concentrates around the mean as the dimension increases

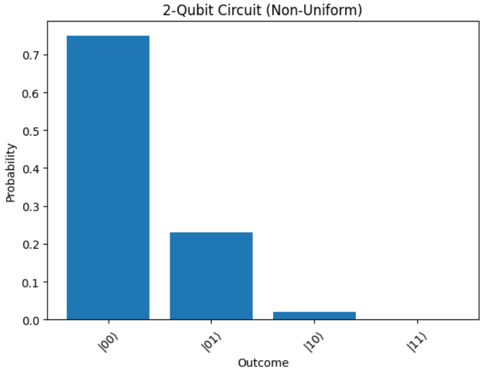
Cybulski, J.L., Nguyen, T., 2023. “Impact of barren plateaus countermeasures on the quantum neural network capacity to learn”, Quantum Inf Processing 22, 442.

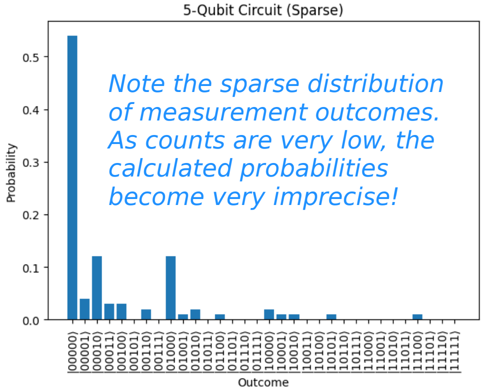
Barren Plateaus (too many parameters)

- Pairwise distances between uniformly distributed points in high-dimensional space become (almost) identical, and the surface of such a space is almost flat.
- In a quantum model with a high-D parameter space, the cost landscape is also flat, the situation called ***barren plateau (BP)***.
- When BPs emerge, the optimiser struggles finding optimum.
- Selecting the optimisation initial point far from the optimum (e.g. random) makes it even more difficult!

Sparse measurement outcomes (too many measurements)

- When increasing the number of measurements,
  we also exponentially increase the number of outcomes, so...
  we need to increase the number of circuit runs exponentially!
- Unless the number of runs is increased with measurements, distribution of outcomes becomes sparse and the probability calculations become imprecise (see fig. below).



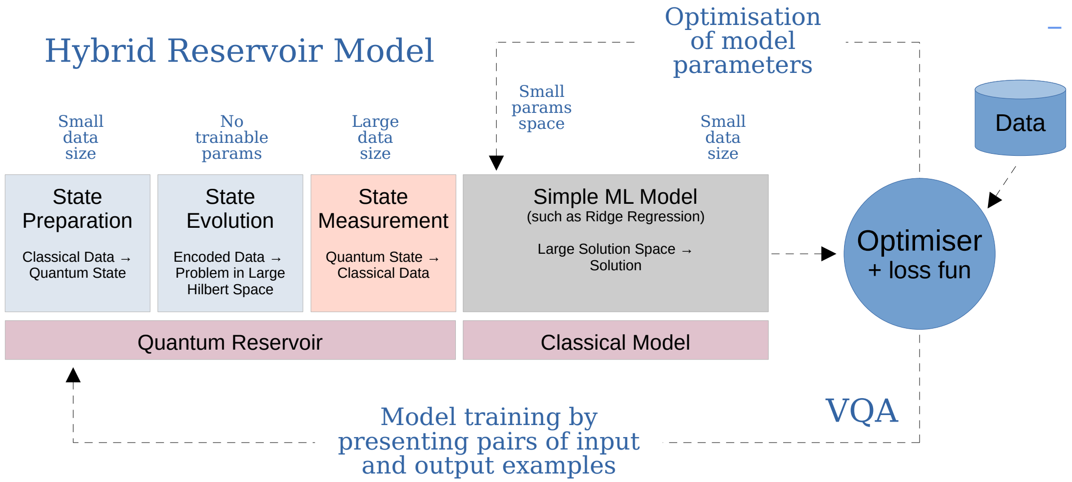


Note the sparse distribution of measurement outcomes. As counts are very low, the calculated probabilities become very imprecise!

Rethinking temporal quantum models

Reduce, Remove, Refine, Rescale, Redesign, Recreate

- **Large TQMs suffer from many ailments, but frugal model design is often a remedy:**
  - reduce number of qubits / layers / parameters
  - remove unnecessary entanglements
  - remove measurements or group them
  - refine parameters initialisation (non-random)
  - use gradient-free optimisers
    (e.g. particle swarm / evolutionary / Bayesian)
  - rescale gradients, adopt quantum-aware optimisation
    (e.g. with Quantum Natural Gradients)



Hybrid Reservoir Model

Optimisation of model parameters

Small params space

Small data size

No trainable params

Large data size

Small data size

Data

State Preparation
Classical Data → Quantum State

State Evolution
Encoded Data → Problem in Large Hilbert Space

State Measurement
Quantum State → Classical Data

Simple ML Model
(such as Ridge Regression)
Large Solution Space → Solution

Optimiser + loss fun

Quantum Reservoir

Classical Model

VQA

Model training by presenting pairs of input and output examples

+ Redesign your model by:

- using BP-resistant models (e.g. QCNNs)
- using BP-resistant model design (e.g. layerwise)
- applying regularisation to loss function
- applying gradient clipping
- using mid-circuit measurements
- using hybrid models (e.g. Hybrid Reservoir Model)

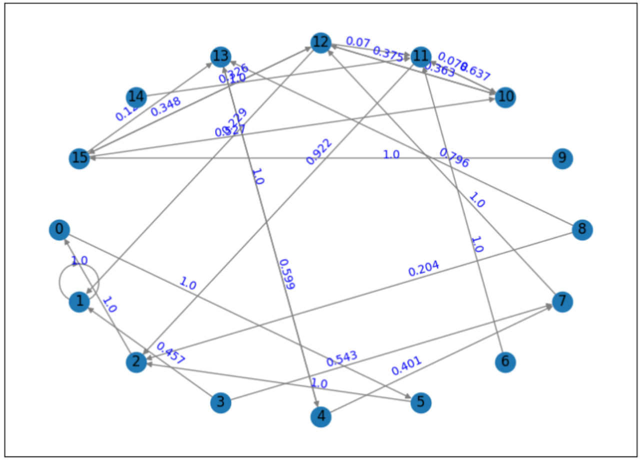
+ If all fails, go analog or classical, consider:

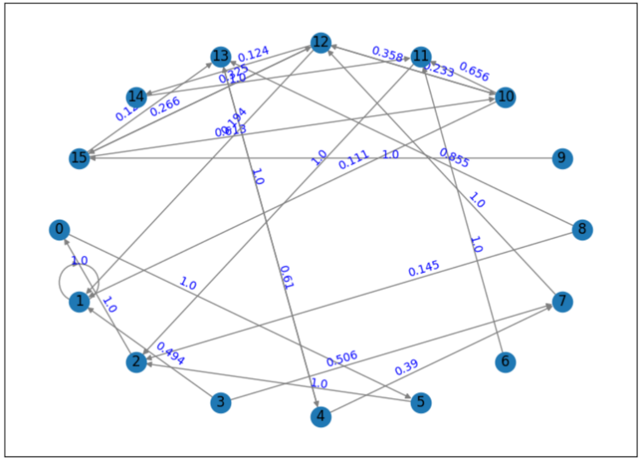
- Analog Rydberg atoms (e.g. Pasqal)
- Quantum annealing (e.g. D-Wave)
- Adiabatic quantum computing (AQC) with Trotterization
- Efficient classical ML approach (e.g. FFT)

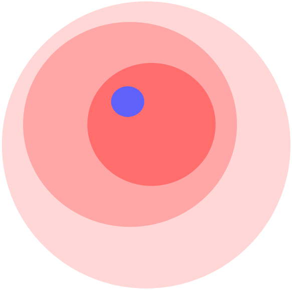
Example of alternative TQM design

- *Reservoir computing is well suited time series analysis*
- *Hybrid quantum reservoir models consist of two parts:*
  - *a quantum reservoir*
  - *a classical model*
- *Quantum reservoir parameters are sparse and randomly assigned – no need for training*
- *The aim of the quantum reservoir is for the model to gain in dimensionality, in order to increase linear separability (similarly to kernel methods)*
- *The classical model (e.g. ridge regression) is simple and easy to train*
- *Overall, the hybrid reservoir model is highly efficient in training + showing high accuracy*

Summary, Current Work and Questions?


Original QGraph


Predicted QGraph (with optimum thr=0.06)



Entangled moments,
Time loops in quantum fabric—
Reality bends.
TQM Haiku by DeepSeek

Dynamic Quantum Graphs

Development of concepts and formalisms related to “quantisation” of classical data structures, such as time series, signals and graphs.

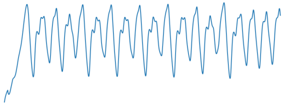
Dynamic quantum graphs for instance will assist highly efficient representation and processing of large, dynamic and highly interconnected structures, e.g. when assisting management of social networks, identification of emerging communities and detection of temporal anomalies in graphs.

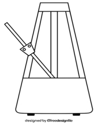
Quantum state evolution

What’s happening in a well-oiled quantum model?

A temporal quantum model can be represented as a quantum circuit, or a hybrid quantum-classical system consisting of quantum circuits and classical computational components, such as ML models, loss functions and optimisers.

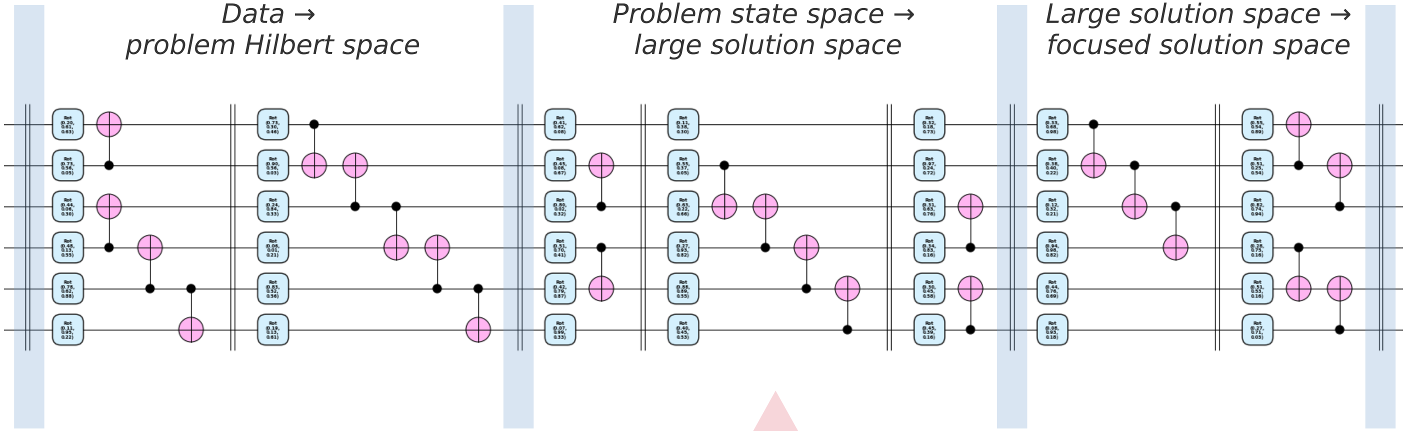
Intended Quantum State Evolution
(data → problem → solutions → outcomes)





State Preparation

Classical Data → Quantum State



Data → problem Hilbert space

Problem state space → large solution space

Large solution space → focused solution space

State Measurement

Quantum State → Classical Data

quantum state accurately represents classical data

quantum state shifts to Hilbert space of ”effective” problem parts and relationships

space of problem states being mapped into space of candidate solution states

quantum state being reduced by less likely solutions cancelling each other

There are no clear boundaries in the quantum state evolution, which similarly to deep learning, takes through the layers of abstractions: from data encoding, problem representation, identification of candidate solutions, highly likely solutions, to measured outcomes.

Quantum advantage happens here

Quantum difficulties also happen here

Measurement and interpretation of the final quantum state is a complex task, which can often be hampered by the quantum technology itself.