

The aims of this session:

*To explore approaches to
quantum time series analysis
and performance of its models*



Introduction

What is expressivity and trainability

Why temporal data is special

QTSA architectures and models

Measuring QTSA model performance

Summary of QTSA performance results

Expectations and surprises

Summary and reflection

Selected readings

Quantum Modelling of Time Series

Expressivity vs. Trainability

Jacob L. Cybulski

Enquanted, Melbourne, Australia

Sebastian Zając

Military University of Technology, Warsaw, Poland

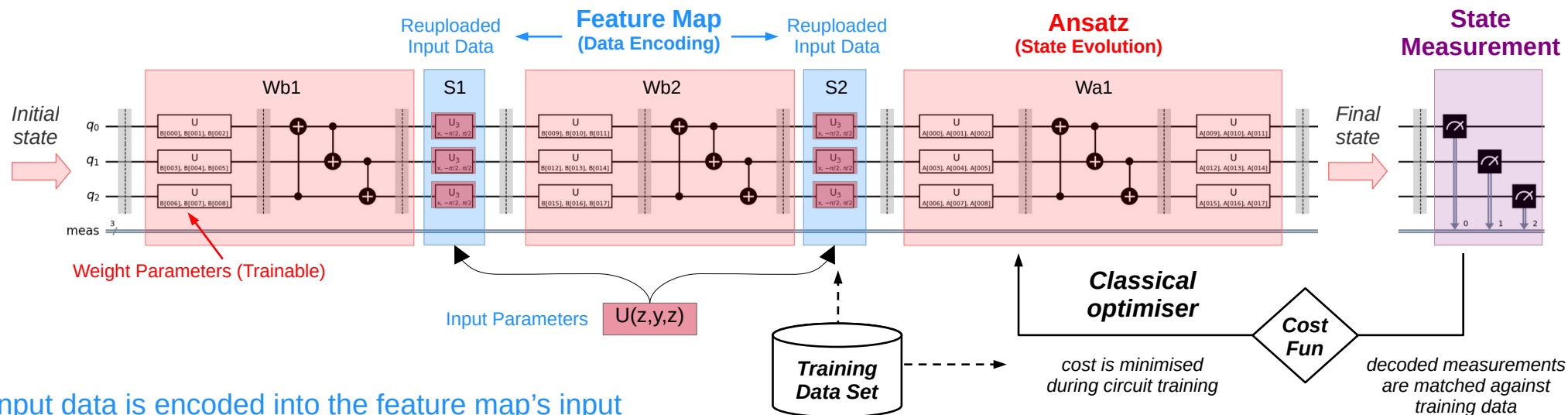
*International Conference on Quantum Computing Infrastructure and Technologies (QUANCOM'2026),
1-4 September, Trento, Italy*



Parameterized Quantum Model (PQC)

Expressivity vs. Trainability

Typically PQCs are trained using Variational Quantum Algorithms (VQA) involving classical optimizers and classical loss / cost functions



Input data is encoded into the feature map's input parameters, setting the initial quantum state and optionally altering states by data reuploading later.

Ansatz evolves the quantum state; it consists of layers of trainable blocks with parameterised quantum gates interwoven with entangling blocks.

The circuit final state is measured and decoded (interpreted) as the model's output in the form of classical data (no mid-circuit measurement).

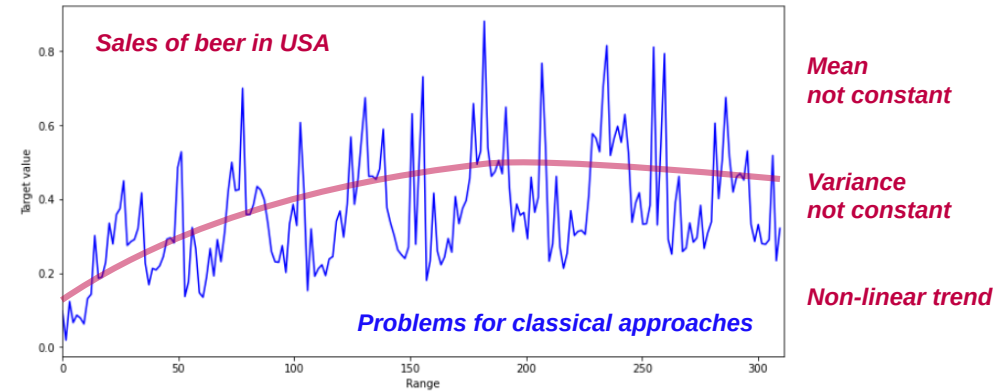
Quantum model effectiveness is influenced by two factors:

- **Expressivity**, i.e. its capacity to effectively represent and process data in quantum space;
- **Trainability**, i.e. its ability to learn and generalise data for predictive tasks and efficiency in the optimisation process.

QTSA

Why quantum in time series (TS)

- Classical approaches to time series analysis are highly effective, i.e., statistical, ML and deep learning!
- However, statistical approaches are restrictive, as they require TS to be well-formed and stationary, i.e., its mean, variance and auto-correlation must be constant
- Quantum time series and signal analysis can help here and were successful in solving problems in many disciplines, e.g.
 - **natural science**, such as medical signal monitoring, earthquake signatures or protein folding;
 - **finance**, such as portfolio optimisation, pricing of share options or detection of market volatility;
 - **optimisation**, such as in energy distribution, adaptive traffic management, or drone control;
 - **machine learning**, such as in radar signal analysis (foe or friend), electric-vehicle navigation, vibration monitoring.
- We consider two types of QTSA models:
 - **Curve-fitting models (CF)**: predict data values at specific time points by fitting a function to the sample data points.
 - **Forecasting models (FC)**: predict future TS values from a sliding window (SW) of their historical context.



- Quantum applications can have advantage over classical solutions by relying on the following features:
 - Randomness of observable results
(measurement of quantum states)
 - Considering alternative decisions concurrently
(superposition of quantum states)
 - Controlling parallel choices with constraints
(entanglement of quantum systems)
- What time series characteristics and optimisation processes can affect quantum model performance?
 - Same issues as in classical analysis, and especially...
 - Auto-correlation of features (temporal dependencies)
 - Feature homogenisation (all values very close)
 - Time series volatility, noise and anomalies
 - Temporal obsolescence (old values not useful), plus...
 - Classical optimiser unaware of quantum phenomena

Methods and Data

- **Methods, for Curve-Fitting and Forecasting models...**

Model types: 3 CF, 3 FC including one classical.

Effectiveness of model types was explored by changing their meta-parameters, e.g. the number of qubits and layers, and their roles.

50 model architectures were generated, i.e.

CF: 5 Serial, 16 Parallel, 5 Xparallel, FC: 14 Xqnn, 6 OvLoad, and 4 Cnn.

Prepared differently instantiated instances (10) of each model.

Implementation in Qiskit was based on EstimatorQNN primitives, trained with NeuralNetworkRegressor, measured expectation values of qubits.

Training with L_BFGS_B optimizer + MSE cost function, and models were scored with MSE and R2 metrics on training and test data partitions.

Reported median+IQR performance of model instances, R2 to compare.

Noise-free simulation was used to avoid confounding factors.

- **Data preparation, for...**

Generated a synthetic time series “2-sins” (simple + repetitions).

Curve-fitting (CF): TS was split into 50 + 20 points for training + testing.

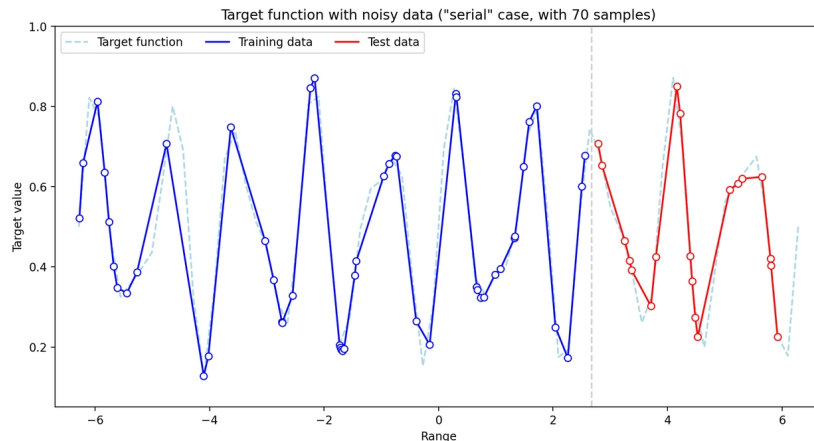
Forecasting (FC): 70 data points formed 65 sliding windows of 5 data points each, offset by 1 step, horizon of 1, incomplete windows discarded. The series was split into 46 + 19 windows for training and testing.

In both cases, training and testing sets had an approximate ratio of 71/29.

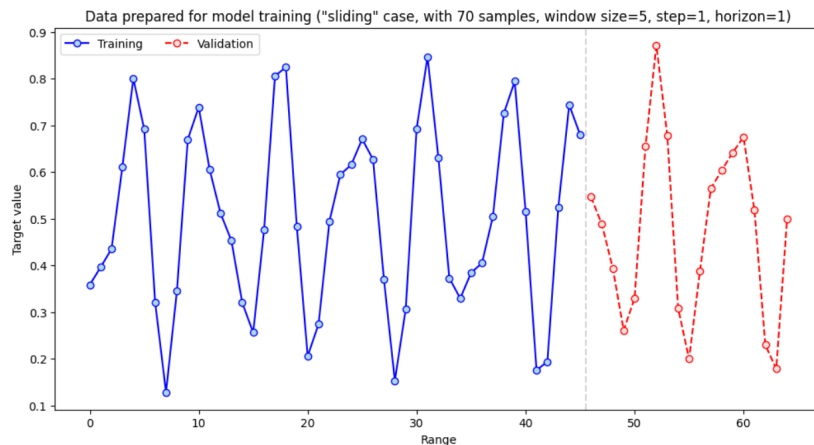
Function 2-sins:

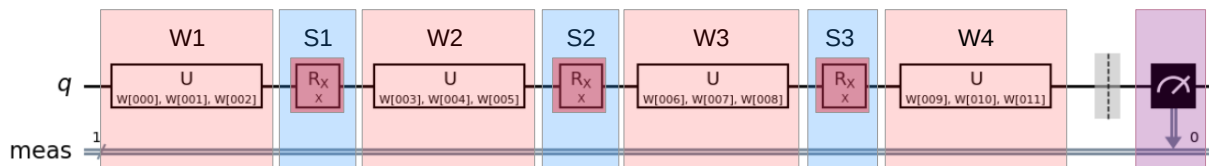
$$f_{2sins}(x) = \frac{\sin(5.0 * x) + 0.5 * \sin(8.0 * x)}{4} + 0.5$$

Function 2-sins sampled for curve-fitting models:



Function 2-sins sampled for sliding-window forecasting models:



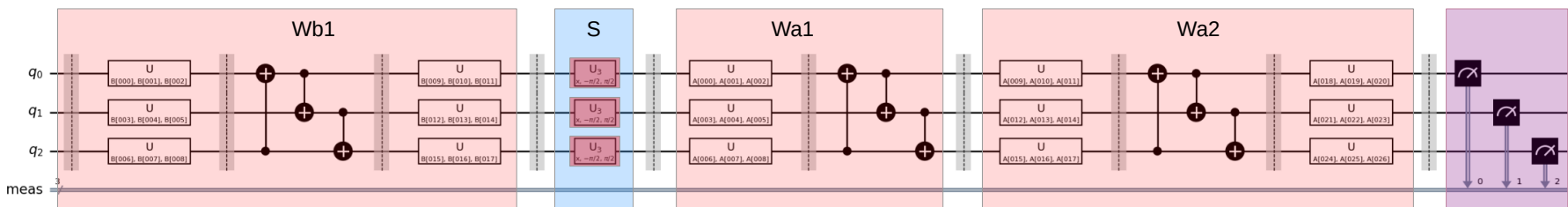


And what we found about them...

CF Serial

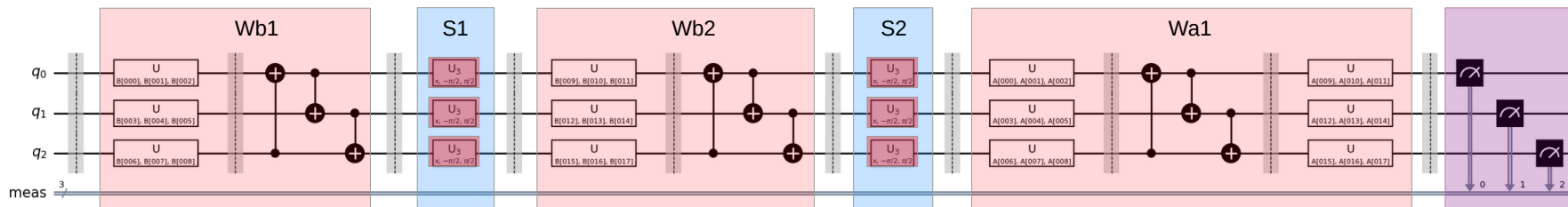
- Serial reuploading + trainable layers → high-dimensional parameter-space → high expressivity in 1 qubit state space
- Hilbert space bounded at 2D → immune to barren plateaus → very fast and effective optimisation!
- Deep circuits, lots of parameters → spectral dilution $\equiv$ thin gradient across Fourier frequency spectrum → unstable tests (IQR)

CF Parallel



- Parallel reuploading + trainable layers → we found they had poor expressivity → little effect of trainable layers
- Training suggested high stability (IQR=0) → instead hard learning capacity ceiling → sub-optimal tests (high IQR)

CF
Extended
Parallel

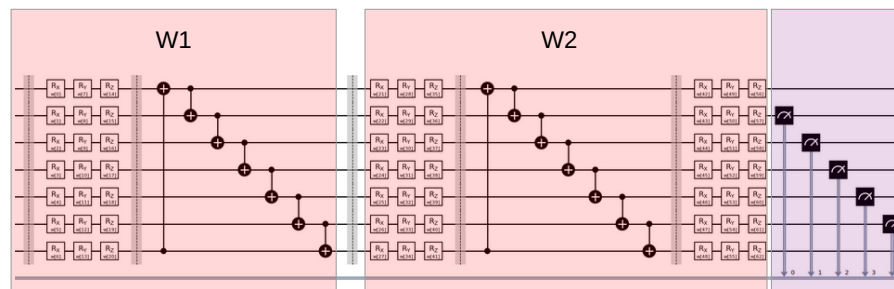
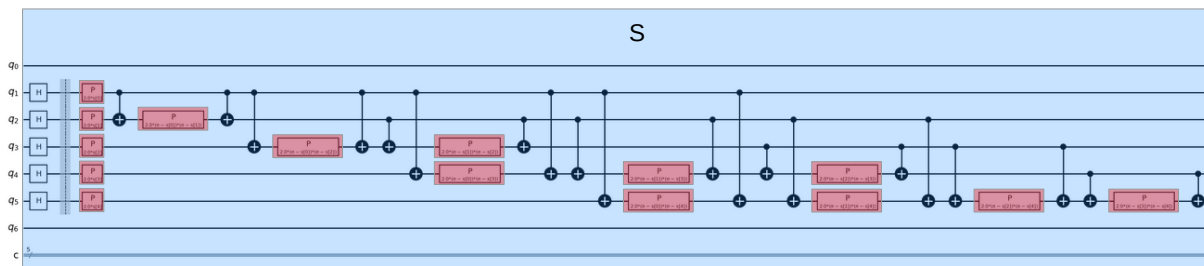


- Parallel + serial reuploading + trainable layers → high expressivity
- Best models: few qubits + deep circuits + not many parameters → superior training and test performance!
- Serial reuploading compensates reduced-dimensionality of the Hilbert space

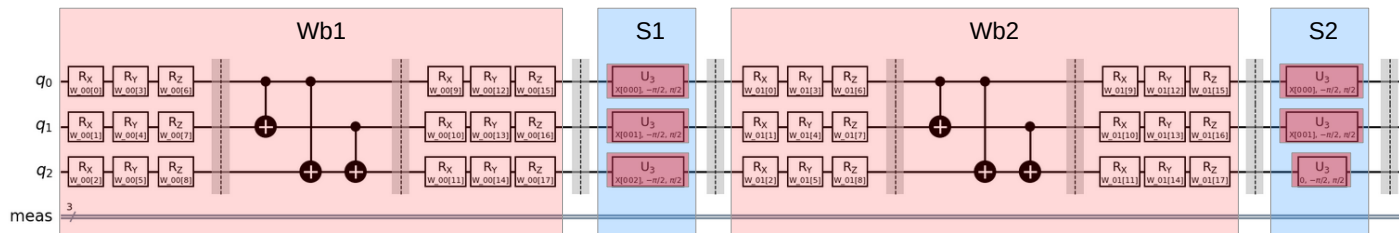
Curve-fitting models

And what we found about them...

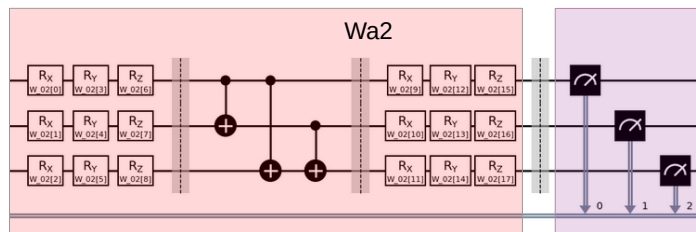
- ZZ feature map → highly expressive data encoding
- Auxiliary qubits → higher-dimensional Hilbert space → more parameters + entanglements → highly expressive model → rapid overfitting
- However, high-dimensional Hilbert space + high entanglement → homogenisation of close TS values → scrambling of closely related states → unstable tests (high IQR).



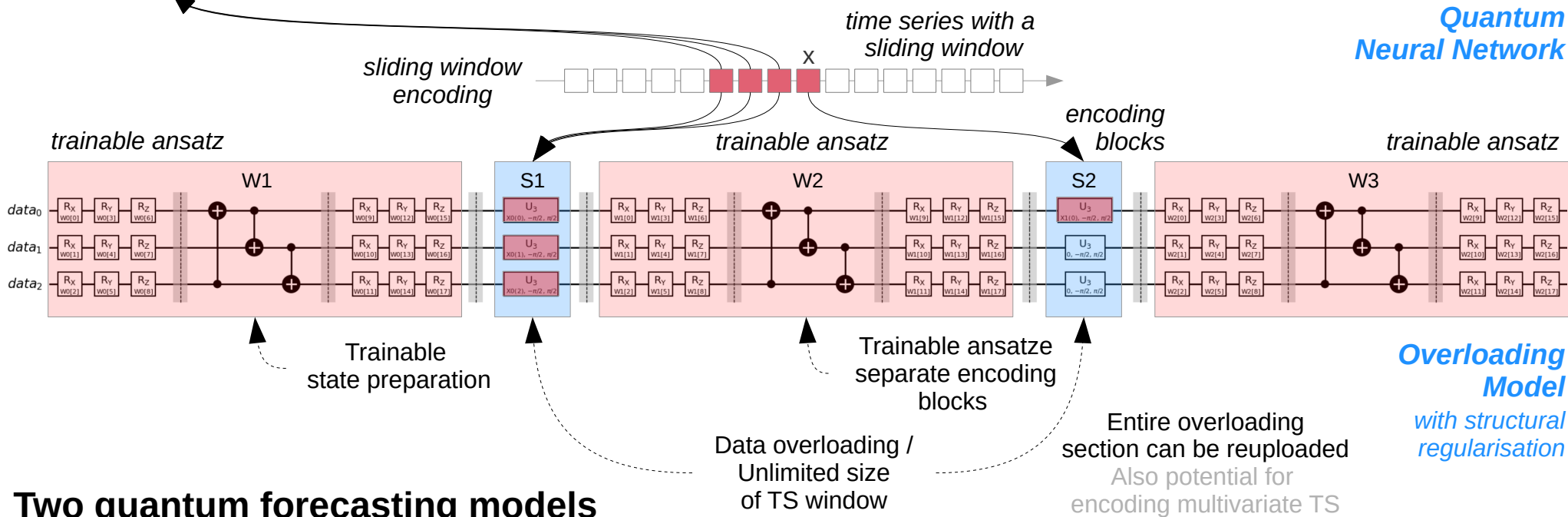
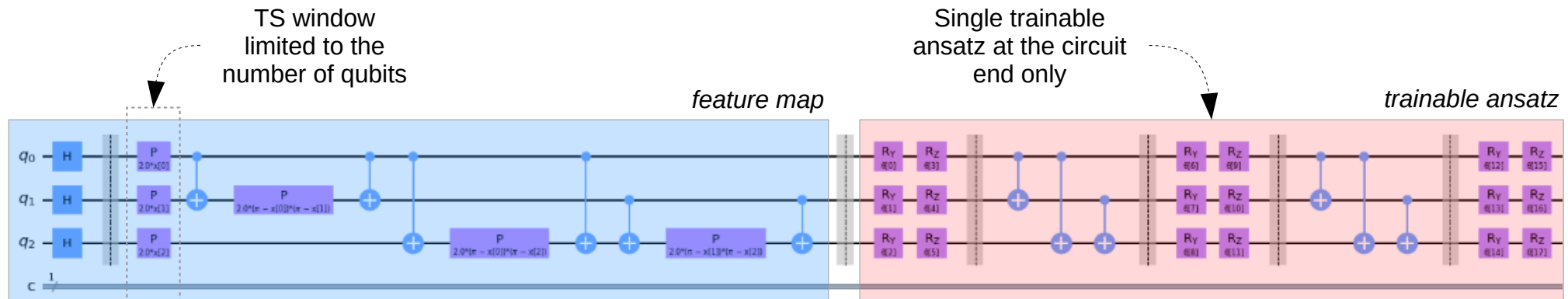
FC Extended QNN
with Havlíček ZZ
feature map



FC Overloading QNN
with structural regularisation

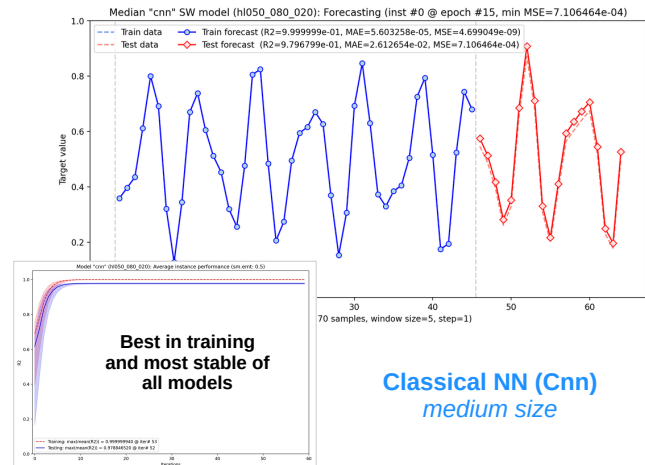
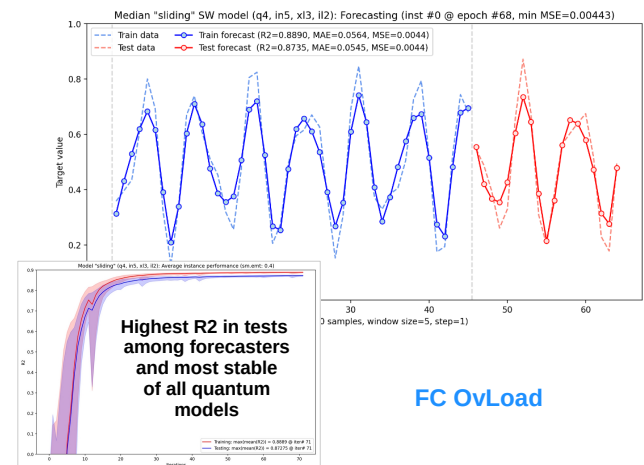
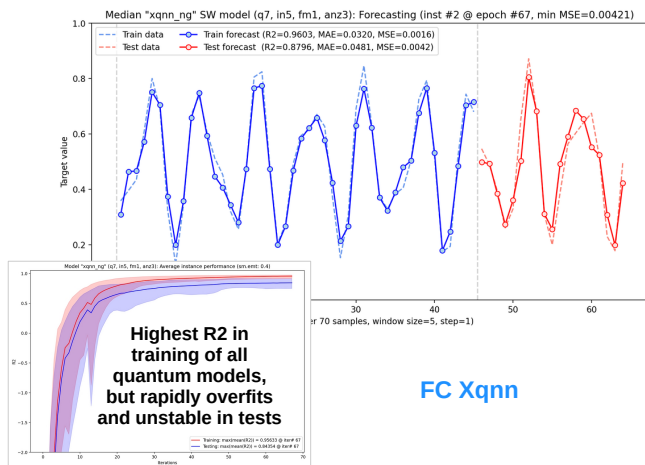
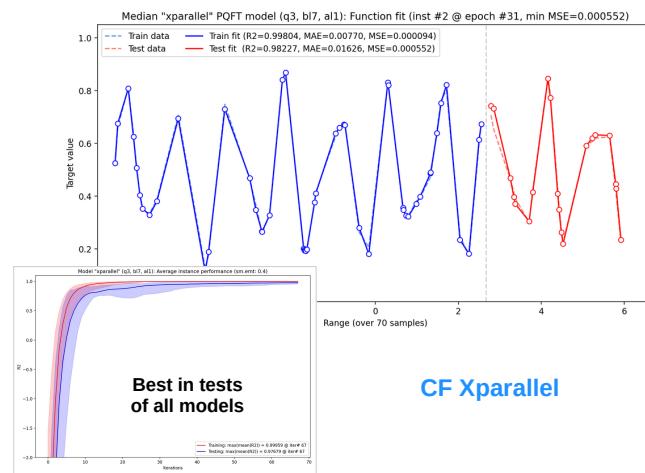
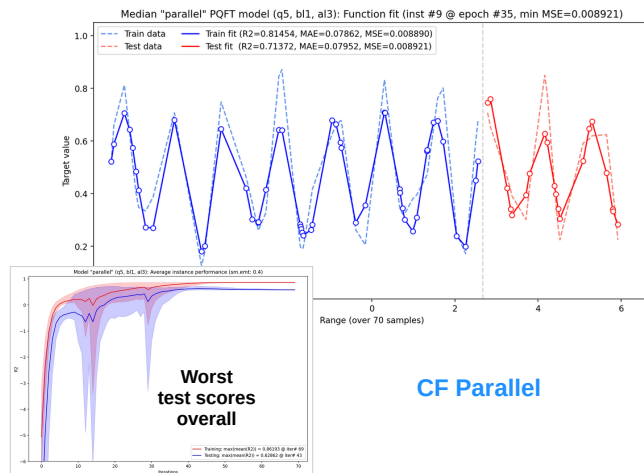
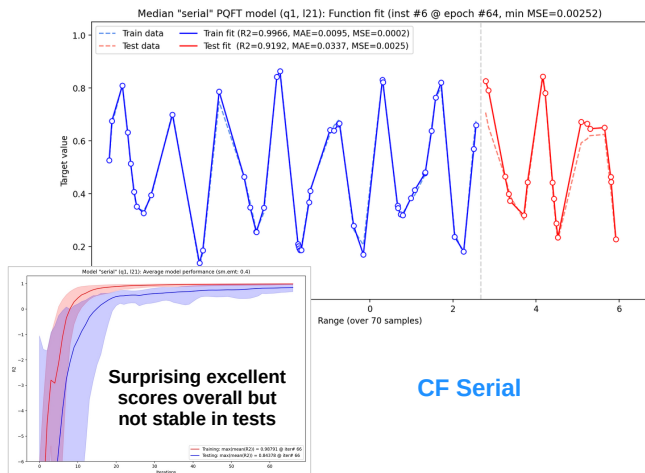


- Overloading of qubit function + serial data reuploading → qubit reuse + highly expressive data encoding
- Prefers high qubits number + high parameters number → structural regularisation → steady performance gain → most stable (low IQR) + highest test performance (R2) (among quantum forecasting models)



Training / Testing, Median R² Scores, and Data Fit

(approximate best configurations selected on test partition shown)



Comparison of different QTSA architectures

Scoring results for the best model variants with median instance performance

Model	Type	Q#	P#	Training R2 (IQR)	Testing R2 (IQR)	Training MSE (IQR)	Testing MSE (IQR)	Specs
serial	CF	1	66	0.9926 (0.0060)	0.9122 (0.1617)	0.0004 (0.0003)	0.0027 (0.0050)	q1 l21
serial	CF	1	84	0.9947 (0.0053)	0.8912 (0.1982)	0.0003 (0.0003)	0.0034 (0.0062)	q1 l27
parallel	CF	5	120	0.8620 (0.0000)	0.6939 (0.0283)	0.0066 (0.0000)	0.0095 (0.0009)	q5 bl3 al3
parallel	CF	5	90	0.8620 (0.0001)	0.7109 (0.0374)	0.0066 (0.0000)	0.0090 (0.0012)	q5 bl1 al3
xparallel	CF	3	81	0.9996 (0.0002)	0.9812 (0.0254)	0.0000 (0.0000)	0.0006 (0.0008)	q3 bl7 al1
xqnn	FC	5	90	0.9789 (0.0087)	0.5607 (0.2022)	0.0008 (0.0003)	0.0154 (0.0071)	q5 in5 fm1 anz5
xqnn	FC	7	84	0.9629 (0.0205)	0.8643 (0.0844)	0.0015 (0.0008)	0.0047 (0.0030)	q7 in5 fm1 anz3
ovload	FC	4	167	0.8888 (0.0013)	0.8734 (0.0018)	0.0044 (0.0001)	0.0044 (0.0001)	q4 xl3 il2
cnn tiny	FC	0	106	1.0000 (0.0021)	0.9684 (0.0420)	0.0000 (0.0001)	0.0011 (0.0015)	hl008 005 003
cnn small	FC	0	375	1.0000 (0.0000)	0.9771 (0.0008)	0.0000 (0.0000)	0.0008 (0.0000)	hl010 015 010
cnn medium	FC	0	5950	1.0000 (0.0000)	0.9796 (0.0010)	0.0000 (0.0000)	0.0007 (0.0000)	hl050 080 020

Note: Q# = Qubit Number, P# = Params Number, CF = Curve-Fitting, FC = Forecasting.

bold = best overall, **red** / **blue** = best training / testing for their model architecture.

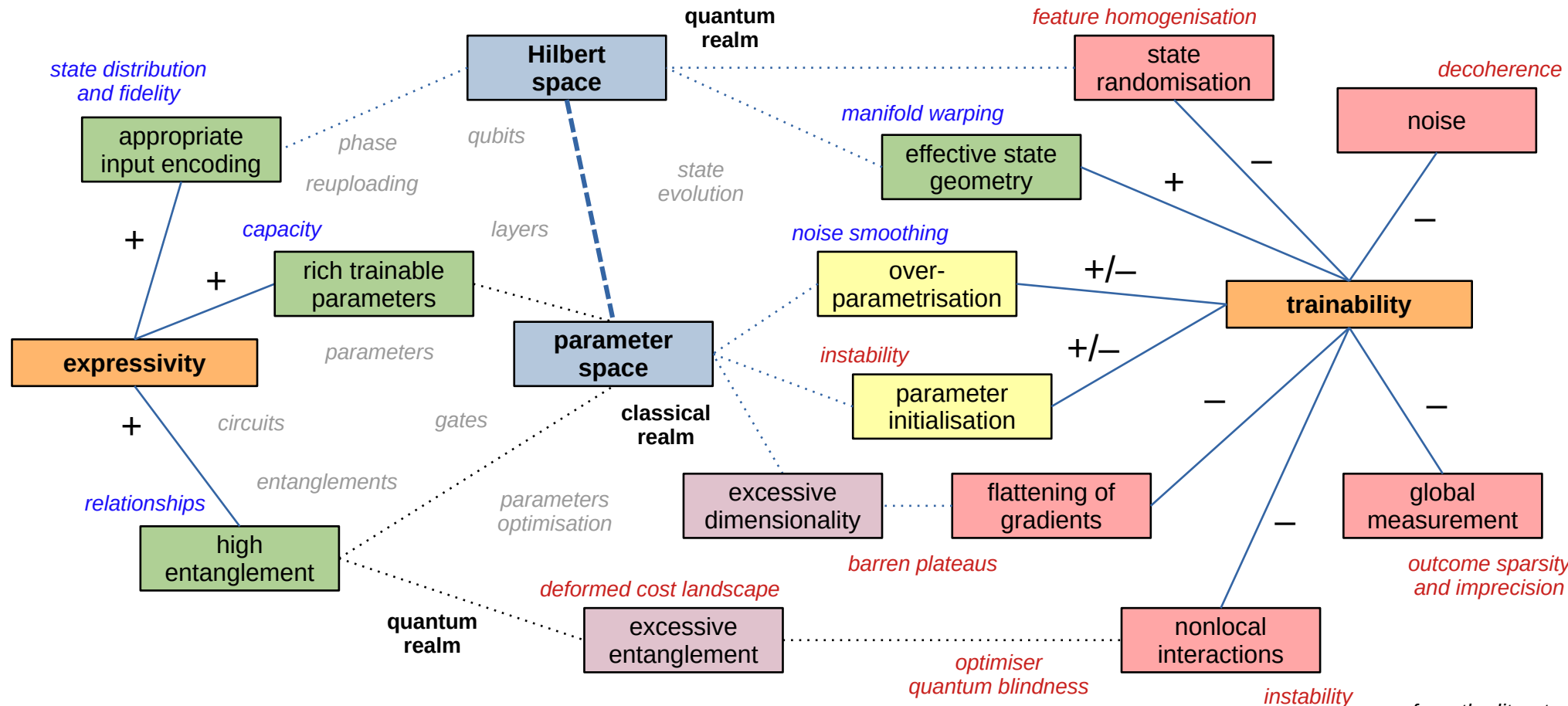
The “Large” Cnn model with 20,800 parameters was excluded from this comparison, as the model had over 100 times more parameters than the largest quantum model.

Expressivity vs. Trainability

lessons learnt for temporal quantum models

Expressivity: capacity to effectively represent and process data in quantum space;

Trainability: ability to learn and generalise data for predictive tasks and efficiency in the optimisation process.



Reflections on QTSA expressivity vs. trainability

- Drawing upon established theoretical principles from prior literature, this talk outlines a plausible architectural blueprint for predictive Quantum Time Series Analysis (QTSA).
- We mapped quantum circuit design principles to the trade-off between model expressivity and trainability.
- By conducting empirical simulations (500 model instances), we confirmed more general QML phenomena and provided new insights about the unique demands of temporal data, to include:
 - **Combating Feature Homogenisation:**
Encoding highly correlated time series into highly entangled circuits risks randomizing the quantum states. We demonstrate that serial data re-uploading is essential to maintain state discernibility in the Hilbert space.
 - **Managing Entanglements:**
Uncontrolled expansion of quantum ansatz depth and width to increase expressivity induces excess entanglements and non-local interactions (confirmed).
Our study maps this directly to temporal tasks, showing (wide IQR) that this flattens the cost landscape and degrades quantum model stability, while narrow, entanglement-controlled circuits outperform wider variants.
 - **Overloading vs. Overparametrisation:**
we introduced overloading (reusing) of qubits with sequential sliding-window data features, to mitigate the trainability penalties of wide circuits and without sacrificing their learning capacity.
Overloading regularises quantum models and smooths the cost landscape without requiring the excessive auxiliary resources linked to the overparametrisation that typically trigger barren plateaus.

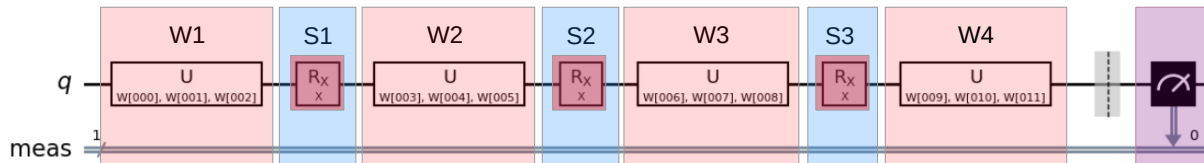
Thank you!
Any questions?



Mathematical model definitions in the Appendix...

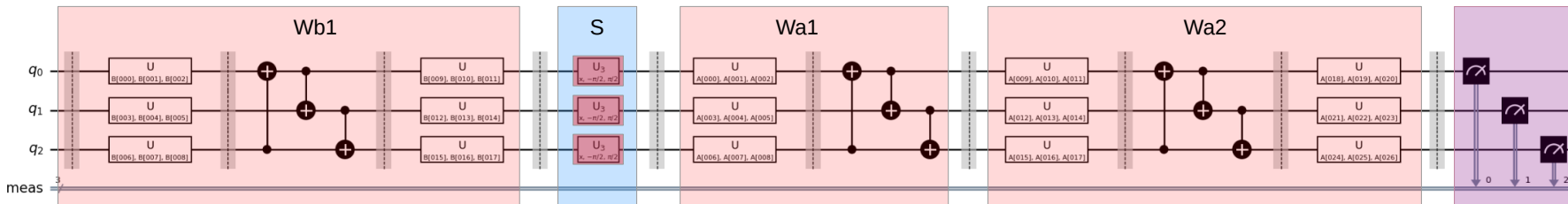
$$|\psi(x, \theta)\rangle = U_{rot}(\theta_{n+1}) \cdot \left(\prod_{j=1}^n [R_X(x) \cdot U_{rot}(\theta_j)] \right) \cdot |0\rangle,$$

CF Serial



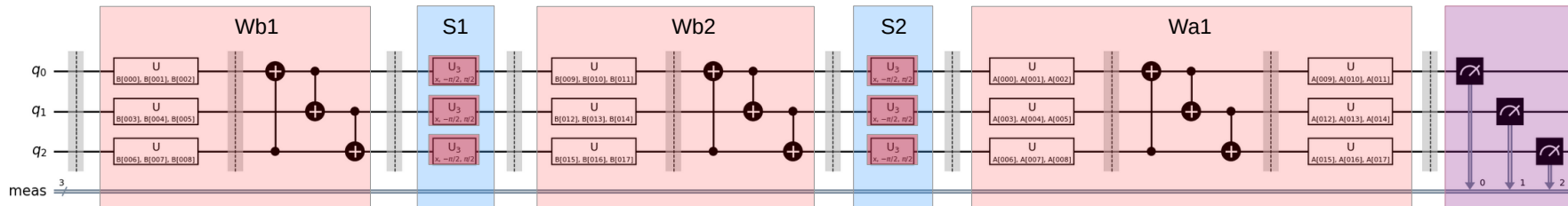
$$|\psi(x, \theta)\rangle = \left(U^q(\theta_{n+1}) \prod_{j=m+2}^n [U_{ent}(\theta_j)] \right) R_X^q(x) \left(U^q(\theta_{m+1}) \prod_{k=1}^m [U_{ent}(\theta_k)] \right) |0\rangle^{\otimes q}$$

CF Parallel



$$|\psi(x, \theta)\rangle = \left(U^q(\theta_{m+1}) \prod_{k=L+1}^m U_{ent}(\theta_k) \right) \cdot \left(\prod_{l=1}^L [R_X^q(x) \cdot U_{ent}(\theta_l)] \right) |0\rangle^{\otimes q}$$

CF Extended Parallel



$$y = \langle 0|^{\otimes q} U^\dagger M U |0\rangle^{\otimes q},$$

$$U(\vec{x}, \vec{\theta}) = U_{ansatz}(\vec{\theta}) \cdot \mathcal{U}_{FM}(\vec{x}),$$

$$U_{ent}(\theta_j) = \prod_{i=1}^q \text{CNOT}_{i, (i \bmod q)+1} \cdot U^q(\theta_j)$$

$$U^q(\theta_j) = \bigotimes_{i=1}^q U_{rot}^{(i)}(\theta_j^{(i)})$$

$$R_X^q(x) = \bigotimes_{j=1}^q R_X^{(j)}(x).$$

Curve-fitting models

$$U_{\Phi(\mathbf{x})} = \exp \left(i \sum_{S \subseteq [n]} \phi_S(\mathbf{x}) \prod_{i \in S} Z_i \right)$$

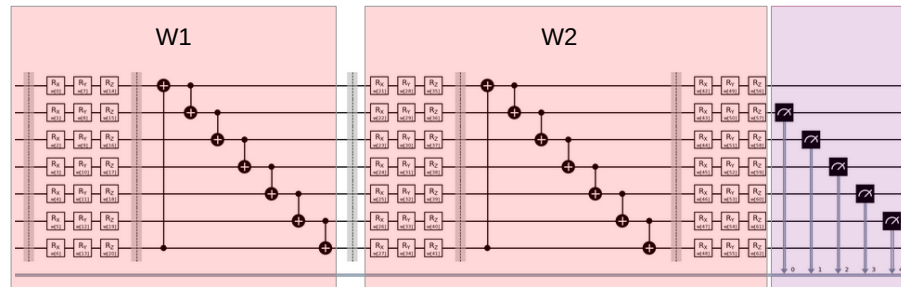
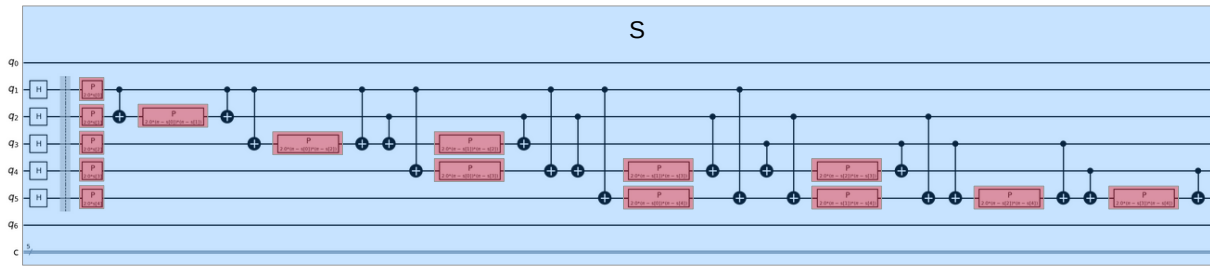
$$\phi_{\{i\}}(\hat{\mathbf{x}}) = x_i$$

$$\phi_{\{i,j\}}(\hat{\mathbf{x}}) = (\pi - x_i)(\pi - x_j)$$

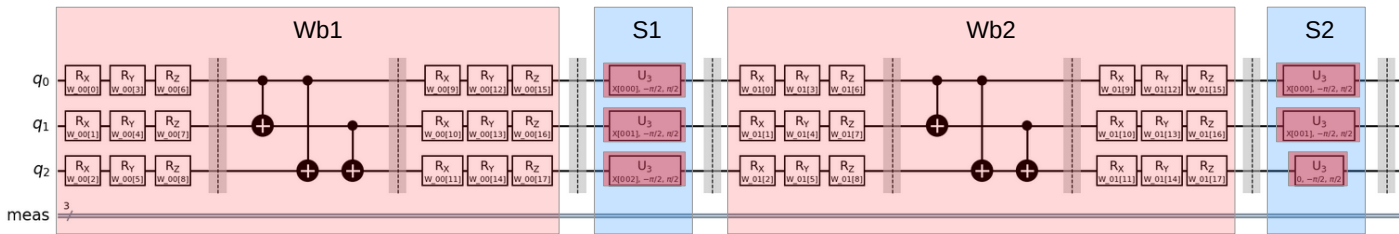
$$\mathcal{U}_{FM}(\mathbf{x}) = U_{\Phi(\mathbf{x})} \cdot H^{\otimes n}$$

$$U_{AN}(\theta) = \prod_{k=1}^{L_{AN}} U_{\text{ent}}(\theta_k)$$

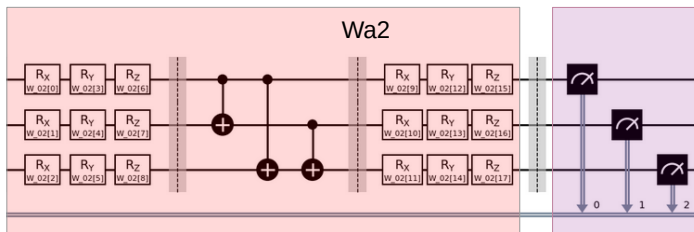
$$|\psi(\mathbf{x}, \theta)\rangle = U_{AN}(\theta) \cdot \mathcal{U}_{FM}(\mathbf{x}) |0\rangle^{\otimes q}$$



FC Extended QNN
with Havlíček ZZ
feature map



FC Overloading QNN
with structural regularisation



$$W_{\text{ent}}^l(\hat{\theta}) = U^q(\theta_{l+1}) \cdot U_{\text{ent}}(\theta_l)$$

$$\mathcal{R}_X^q(\hat{\mathbf{x}}) = \prod_{k=0}^{\lceil n/q \rceil - 1} \left(\bigotimes_{j=1}^q R_X^{(j)}(x_{kq+j}) \right)$$

$$|\psi(\hat{\mathbf{x}}, \hat{\theta})\rangle = W_{\text{ent}}^{L+1}(\hat{\theta}) \cdot \left(\prod_{k=1}^L [\mathcal{R}_X^q(\hat{\mathbf{x}}) \cdot W_{\text{ent}}^k(\hat{\theta})] \right) |0\rangle^{\otimes q}$$