

Supplement

This is a supplement for the paper by Jacob L. Cybulski and Sebastian Zajac, "Quantum Modelling of Time Series: Expressivity vs. Trainability", Proc. Int. Conf. on Quantum Computing Infrastructure and Technologies (QUANCOM 2026), 1-4 September, Trento, Italy.

The following figures and tables explain the model structures and summarize performance of quantum models of two categories (Type column), i.e. curve-fitting (CF) and forecasting (FC). Within each category, we distinguish between different model architectures (Model column), which were tested by varying their properties (Specs column), such as the number of qubits, as well as, types and the number of layers (within the encoding and ansatz blocks).

- **Curve-fitting models** predict data values at specific timestamps by fitting a function to the sample data points. We evaluate three data re-uploading strategies in the following model architectures: single-qubit *serial* (Figure A and Table A), multi-qubit *parallel* (Figure B and Table B), and extended multi-qubit *xparallel* (Figure C and Table C).
- **Forecasting models** predict future TS values from a sliding window (SW) of their historical context. We evaluate two architectural variants: *xqnn*, which distributes the input window across dedicated qubits while extending circuit width for overparameterization (Figure D and Table D), and *ovload*, which overcomes circuit-width constraints by splitting long temporal windows into fragments encoded sequentially between trainable layers via qubit reuse (Figure E and Table E). A classical neural network (*cnn*) utilizing identical sliding-window inputs was also included to serve as the baseline benchmark (Table F).

At the very end, we compare the best models within each model architecture (Table G).

As model variants are created (in Qiskit), trained (using L_BFGS_B optimizer, over 60 epochs, with MSE as objective function) and tested (using MSE and R^2 scores) on 10 differently initialized instances, each execution resulted in different level of performance. For each variant, we selected the median performing instance as the representative of that architectural variant, and indicate the performance spread using the Interquartile Range (IQR). As model categories differ in the way they use training data and the way they interpret errors, the MSE metrics can only be used to compare the models with their own category. However R^2 measures are applicable to all models, and therefore is used as the primary assessment of their training and test performance.

In all tables, the best performance scores have been shown in bold face, to indicate the minimum for MSE and maximum for R^2 . On some occasions, several models achieved exactly the same result, in which case all scores have been highlighted. In other situations, seemingly identical results were not equally highlighted, which is due to their small performance differences obscured by display precision to 4 decimal places - those results that have not been placed in bold were not the best.

1 Serial models

The curve-fitting *serial* models were unique, since they used only a single qubit (q1), with several layers (l) of data reuploading and trainable blocks.

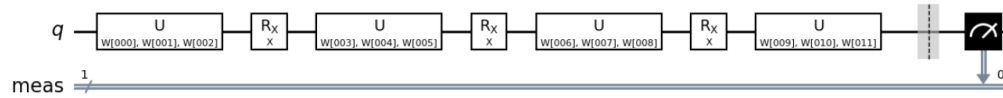


Figure A: Sample serial model (Arch specs: q1 l3)

Table A: Comparison of model "serial" performance (data="2 sins")

Model	Type	Qubits	Params	Training R2 (IQR)	Testing R2 (IQR)	Training MSE (IQR)	Testing MSE (IQR)	Specs
serial	CF	1	12	0.1087 (0.0082)	-0.1338 (0.1434)	0.0427 (0.0004)	0.0353 (0.0045)	q1 l3
serial	CF	1	30	0.8892 (0.0580)	0.5120 (0.0833)	0.0053 (0.0028)	0.0152 (0.0026)	q1 l9
serial	CF	1	48	0.9835 (0.0396)	0.7505 (0.2510)	0.0008 (0.0019)	0.0078 (0.0078)	q1 l15
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

2 Parallel models

In experiments with curve-fitting *parallel* models, the circuit architectures were varied by changing the number of qubits (q), layers of trainable pre-encoding blocks (bl) and layers of trainable post-encoding blocks (al).

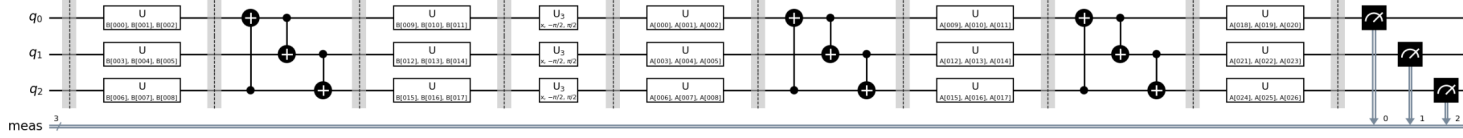


Figure B: Sample parallel model (Arch specs: q3 bl1 al2)

Table B: Comparison of model "parallel" performance (data="2 sins")

Model	Type	Qubits	Params	Training R2 (IQR)	Testing R2 (IQR)	Training MSE (IQR)	Testing MSE (IQR)	Specs
parallel	CF	2	48	0.0588 (0.0000)	-0.1407 (0.0391)	0.0451 (0.0000)	0.0355 (0.0012)	q2 bl3 al3
parallel	CF	3	72	0.1088 (0.0000)	-0.1652 (0.1275)	0.0427 (0.0000)	0.0363 (0.0040)	q3 bl3 al3
parallel	CF	4	96	0.1096 (0.0000)	-0.2544 (0.1597)	0.0427 (0.0000)	0.0391 (0.0050)	q4 bl3 al3
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	60	0.8604 (0.0082)	0.7079 (0.0297)	0.0067 (0.0004)	0.0091 (0.0009)	q5 bl1 al1
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	180	0.8620 (0.0000)	0.6816 (0.0351)	0.0066 (0.0000)	0.0099 (0.0011)	q5 bl5 al5
parallel	CF	5	240	0.8620 (0.0000)	0.6690 (0.0310)	0.0066 (0.0000)	0.0103 (0.0010)	q5 bl7 al7
parallel	CF	5	60	0.8604 (0.0082)	0.7079 (0.0297)	0.0067 (0.0004)	0.0091 (0.0009)	q5 bl1 al1
parallel	CF	5	90	0.8620 (0.0001)	0.7109 (0.0374)	0.0066 (0.0000)	0.0090 (0.0012)	q5 bl1 al3
parallel	CF	5	120	0.8620 (0.0000)	0.6959 (0.0240)	0.0066 (0.0000)	0.0095 (0.0007)	q5 bl1 al5
parallel	CF	5	150	0.8620 (0.0000)	0.6989 (0.0249)	0.0066 (0.0000)	0.0094 (0.0008)	q5 bl1 al7
parallel	CF	5	90	0.8620 (0.0001)	0.7109 (0.0374)	0.0066 (0.0000)	0.0090 (0.0012)	q5 bl1 al3
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	150	0.8620 (0.0000)	0.7013 (0.0246)	0.0066 (0.0000)	0.0093 (0.0008)	q5 bl5 al3
parallel	CF	5	180	0.8620 (0.0000)	0.6877 (0.0299)	0.0066 (0.0000)	0.0097 (0.0009)	q5 bl7 al3

3 Xparallel models

Xparallel curve-fitting models were designed to extend the *parallel* models by adding sequential data reuploading. Experiments with these multi-qubit models focused on varying the number of qubits (q) and layers consisting of pairs: parallel reuploading blocks and parametrized training blocks (bl). The sequence of entangling and reuploading layers ended with optional parametrized entangling blocks (al, limited to 1). All models had a similar number of trainable parameters (69-84).

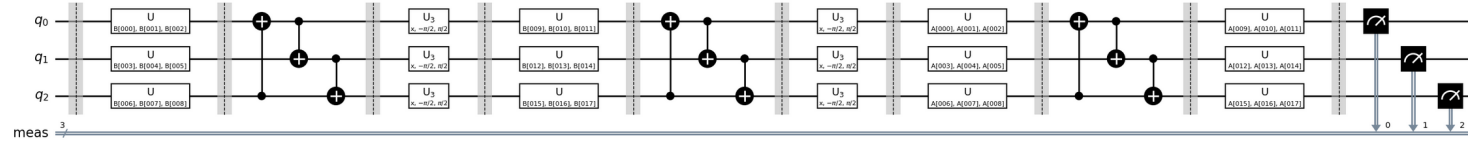


Figure C: Sample xparallel model (Arch specs: q3 bl2 al1)

Table C: Comparison of model "xparallel" performance (data="2 sins")

Model	Type	Qubits	Params	Training R2 (IQR)	Testing R2 (IQR)	Training MSE (IQR)	Testing MSE (IQR)	Specs
xparallel	CF	1	69	0.9963 (0.0072)	0.9434 (0.0571)	0.0002 (0.0003)	0.0018 (0.0018)	q1 bl21 al1
xparallel	CF	3	81	0.9996 (0.0002)	0.9812 (0.0254)	0.0000 (0.0000)	0.0006 (0.0008)	q3 bl7 al1
xparallel	CF	5	75	0.9678 (0.0152)	0.9031 (0.0264)	0.0015 (0.0007)	0.0030 (0.0008)	q5 bl3 al1
xparallel	CF	7	84	0.8253 (0.1147)	0.7168 (0.1058)	0.0084 (0.0055)	0.0088 (0.0033)	q7 bl2 al1
xparallel	CF	9	81	0.6801 (0.2870)	0.5810 (0.3647)	0.0153 (0.0138)	0.0131 (0.0114)	q9 bl1 al1

4 Xqnn models

Xqnn models provide a small extension to the traditional QNNs to allow the addition of auxiliary qubits, consequently resulting in the widening of circuits, thus gaining more rotational parameters and entanglements. The model encoding adopted a traditional multilayered ZZ feature map (fm), embedding windows of 5 input values (in), followed by a post-encoding layer of several trainable blocks (anz), with optional auxiliary qubits (to a total of q qubits). For consistency, only qubits encoding input were measured. The experiments with *xqnn* models aimed to investigate their capacity to overparameterize, which also increased the model entanglement.

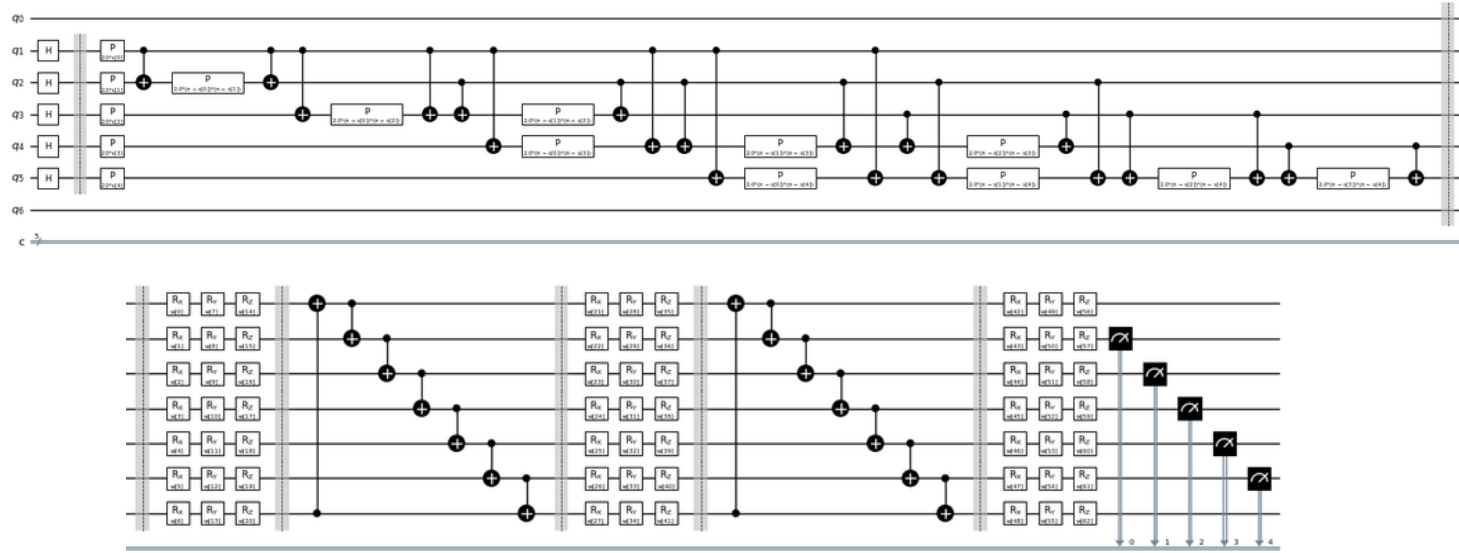


Figure D: Sample xqnn model (Arch specs: q7 in5 fm1 anz2)

Table D: Comparison of model "sw xqnn" performance (data="2 sins")

Model	Type	Qubits	Params	Training R2 (IQR)	Testing R2 (IQR)	Training MSE (IQR)	Testing MSE (IQR)	Specs
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	6	108	0.9353 (0.0536)	0.6676 (0.1303)	0.0025 (0.0021)	0.0116 (0.0046)	q6 in5 fm1 anz5
xqnn	FC	7	126	0.9772 (0.0177)	0.8358 (0.0707)	0.0009 (0.0007)	0.0057 (0.0025)	q7 in5 fm1 anz5
xqnn	FC	8	144	0.9366 (0.0567)	0.8113 (0.0727)	0.0025 (0.0022)	0.0066 (0.0025)	q8 in5 fm1 anz5
xqnn	FC	9	162	0.9655 (0.0278)	0.8437 (0.0799)	0.0014 (0.0011)	0.0055 (0.0028)	q9 in5 fm1 anz5
xqnn	FC	7	42	0.8946 (0.0458)	0.8410 (0.0581)	0.0042 (0.0018)	0.0056 (0.0020)	q7 in5 fm1 anz1
xqnn	FC	7	63	0.9579 (0.0180)	0.8535 (0.0766)	0.0017 (0.0007)	0.0051 (0.0027)	q7 in5 fm1 anz2
xqnn	FC	7	84	0.9629 (0.0205)	0.8643 (0.0844)	0.0015 (0.0008)	0.0047 (0.0030)	q7 in5 fm1 anz3
xqnn	FC	7	105	0.9665 (0.0318)	0.8456 (0.0994)	0.0013 (0.0013)	0.0054 (0.0035)	q7 in5 fm1 anz4
xqnn	FC	7	126	0.9772 (0.0177)	0.8358 (0.0707)	0.0009 (0.0007)	0.0057 (0.0025)	q7 in5 fm1 anz5
xqnn	FC	7	147	0.8928 (0.1596)	0.7310 (0.3260)	0.0042 (0.0063)	0.0094 (0.0114)	q7 in5 fm1 anz6
xqnn	FC	7	84	0.9629 (0.0205)	0.8643 (0.0844)	0.0015 (0.0008)	0.0047 (0.0030)	q7 in5 fm1 anz3
xqnn	FC	7	84	0.8505 (0.0314)	0.5829 (0.0886)	0.0059 (0.0012)	0.0146 (0.0031)	q7 in5 fm2 anz3
xqnn	FC	7	84	0.8705 (0.0210)	0.2569 (0.1413)	0.0051 (0.0008)	0.0260 (0.0049)	q7 in5 fm3 anz3

5 Ovload models

Ovload models adopt the concept of multi-qubit data reuploading technique from the curve-fitting *xparallel* models. However, they tailor it to represent sliding TS windows, rather than individual TS values, to reduce the demand for qubit resources. This is accomplished by splitting longer windows into a number of smaller fragments (il) that fit the available qubits (q). The *ovload* layers (xl) are then formed by encoding all window fragments and interleaving them with trainable blocks. As a result, the circuit's qubits are overloading the representational function for the windows sub-sequences, which relies on the continuity of reuploaded consecutive data points in a TS window.

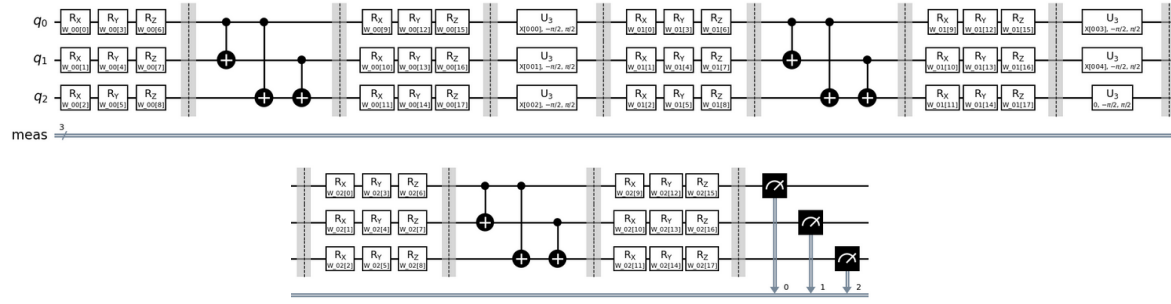


Figure E: Sample overloading model (Arch specs: q3 in5 xl1 il2)

Table E: Comparison of model "sw ovload" performance (data="2 sins")

Model	Type	Qubits	Params	Training R2 (IQR)	Testing R2 (IQR)	Training MSE (IQR)	Testing MSE (IQR)	Specs
ovload	FC	1	92	0.2321 (0.0000)	0.1557 (0.0036)	0.0302 (0.0000)	0.0295 (0.0001)	q1 xl3 il5
ovload	FC	2	81	0.7856 (0.0000)	0.7697 (0.0013)	0.0084 (0.0000)	0.0081 (0.0000)	q2 xl2 il3
ovload	FC	3	88	0.8726 (0.0010)	0.8501 (0.0018)	0.0050 (0.0000)	0.0052 (0.0001)	q3 xl2 il2
ovload	FC	4	71	0.8848 (0.0021)	0.8663 (0.0021)	0.0045 (0.0001)	0.0047 (0.0001)	q4 xl1 il2
ovload	FC	4	119	0.8878 (0.0014)	0.8723 (0.0011)	0.0044 (0.0001)	0.0045 (0.0000)	q4 xl2 il2
ovload	FC	4	167	0.8888 (0.0013)	0.8734 (0.0018)	0.0044 (0.0001)	0.0044 (0.0001)	q4 xl3 il2

6 Classical QNN models

Experiments with classical *cnn* neural networks were included to provide a reference to compare with the performance of the remaining quantum models. All *cnn* models had three hidden layers with varying numbers of parameters. Four models were included, the “tiny” model (131 parameters), “small” (426 parameters), “medium” (6181 parameters) and “large” (20800 parameters).

Table F: Comparison of model "sw cnn" performance (data="2 sins")

Model	Type	Qubits	Params	Training R2 (IQR)	Testing R2 (IQR)	Training MSE (IQR)	Testing MSE (IQR)	Specs
cnn	FC	0	106	1.0000 (0.0021)	0.9684 (0.0420)	0.0000 (0.0001)	0.0011 (0.0015)	hl008 005 003
cnn	FC	0	375	1.0000 (0.0000)	0.9771 (0.0008)	0.0000 (0.0000)	0.0008 (0.0000)	hl010 015 010
cnn	FC	0	5950	1.0000 (0.0000)	0.9796 (0.0010)	0.0000 (0.0000)	0.0007 (0.0000)	hl050 080 020
cnn	FC	0	20800	1.0000 (0.0000)	1.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	hl150 100 050

As expected, the classical models exhibited vastly superior trainability and stability, consistently achieving near-perfect training scores regardless of initialization. However, the quantum models demonstrated a notable advantage in parameter efficiency, while approaching and in some cases marginally exceeding the performance of small-to-medium size classical models.

7 Comparison of models

The following table summarizes the best results for the variants in each model category. For some model types, the same model parameterization resulted in the optimum for both training and testing, in which case a single model variant, shown as a single table row, is shown to represent the model type. In other cases, two variants ended up as best in training separately from testing, and so two variants, shown in two table rows, represent the model type.

Note that the massive classical model of 20800 parameters was not included in this table as it did not compare well with the quantum models size-wise.

Table G: Comparison of different QTSA architectures (data="2 sins")

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	FC	0	106	1.0000 (0.0021)	0.9684 (0.0420)	0.0000 (0.0001)	0.0011 (0.0015)	hl008 005 003
cnn	FC	0	375	1.0000 (0.0000)	0.9771 (0.0008)	0.0000 (0.0000)	0.0008 (0.0000)	hl010 015 010
cnn	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.

The analysis reviewed five types of QTSA models within the curve-fitting and forecasting model groups. It evaluated their architectural variants and their performance based on their design. Most importantly, it established the connection of model design to its expressivity and trainability. The complete QUNACOM'2026 paper provides discussion of these results and a synthesis of the insights derived from this research.