arXiv:2404.15377v2 [quant-ph] 25 Apr 2024

Fourier Series Guided Design of Quantum Convolutional Neural Networks for Enhanced Time Series Forecasting

Sandra Leticia Juárez Osorio[1][0009−0003−5525−1309], Mayra Alejandra Rivera Ruiz[1][0000−0002−2660−1520], Andres Mendez-Vazquez[1][0000−0001−7121−8195], and Eduardo Rodriguez-Tello[2][0000−0002−0333−0633]

[1] CINVESTAV Unidad Guadalajara, Av. del Bosque 1145, Colonia el Bajío, C.P, 45019, Zapopan, Jalisco, México
[2] CINVESTAV Unidad Tamaulipas, Km. 5.5 Carretera Victoria - Soto La Marina, C.P, 87130, Victoria, Tamaulipas, Mexico
{sandra.juarez,mayra.rivera,andres.mendez,ertello}@cinvestav.mx

Abstract. In this study, we apply 1D quantum convolution to address the task of time series forecasting. By encoding multiple points into the quantum circuit to predict subsequent data, each point becomes a feature, transforming the problem into a multidimensional one. Building on theoretical foundations from prior research, which demonstrated that Variational Quantum Circuits (VQCs) can be expressed as multidimensional Fourier series, we explore the capabilities of different architectures and ansatz. This analysis considers the concepts of circuit expressibility and the presence of barren plateaus. Analyzing the problem within the framework of the Fourier series enabled the design of an architecture that incorporates data reuploading, resulting in enhanced performance. Rather than a strict requirement for the number of free parameters to exceed the degrees of freedom of the Fourier series, our findings suggest that even a limited number of parameters can produce Fourier functions of higher degrees. This highlights the remarkable expressive power of quantum circuits. This observation is also significant in reducing training times. The ansatz with greater expressibility and number of non-zero Fourier coefficients consistently delivers favorable results across different scenarios, with performance metrics improving as the number of qubits increases.

1 Introduction

Today, Quantum Computing (QC) is still in the Noisy intermidiate-Scale Quantum (NISQ) era: devices with a small number of qubits and errors. Variational Quantum Circuits work with few qubits, being suitable to this kind of devices [5]. These VQCs are capable of working in conjunction with classical layers in order to divide the task between a classical and a quantum computer. Several applications to these hybrid algorithms can be found in the literature. In this

work, the focus is on quantum machine learning, specifically in the utilization of Variational Quantum Circuits (VQCs) to construct a quantum version of the widely used Convolutional Neural Networks (CNN's).

In the literature, numerous instances abound where Quantum Machine Learning (QML) algorithms have demonstrated superior or comparable performance to their classical counterparts. This holds true across various scenarios, encompassing toy datasets ([21], [16], [13]) as well as real-world applications, such as medical image classification ([19], [11], [24]), defects detection in materials [27], and time series forecasting ([2], [18]). Notably, Quantum Convolutional Neural Networks (QCNNs) have found application not only in toy datasets like MNIST and Fashion MNIST ([12], [8]) but also in practical contexts, such as protein distance prediction [10] and the identification of COVID-19-infected patients [11].

While some of these works adopt an empirical approach to achieve superior results compared to classical counterparts, our focus in this study is to provide insights based on existing theory, specifically drawing from [23], [4], [25], and [9]. The theoretical foundation described in those works is helpful to establish a general framework to design an efficient architecture that yields both good metrics and manageable training times.

In [23] and [4], a demonstration of how VQCs can be viewed as Fourier Series (FS) is depicted, establishing a relationship between the number of layers, data reuploading, and the expressibility of the circuit. [25] introduces a measure of ansatz expressibility, while [9] discusses the challenge of training circuits with flat cost landscapes. This work seeks to empirically validate the impact of these parameters when training with real data, utilizing them to design an effective architecture.

The objective is not to prescribe a definitive architecture but to provide key insights derived from the theories of these foundational works. Focusing specifically on time series forecasting, the study acknowledges that training times for large datasets can be prohibitive. However, the findings offer a starting point for designing architectures that may be applied in future research with similar, albeit larger, datasets.

2 Background and related work

2.1 Quantum Computing

The emergence of quantum computers, proposed by Feynman in 1982 for simulating quantum systems, has evolved significantly in the past four decades, despite enduring accuracy limitations in quantum processors [7,17]. Quantum algorithms have showcased supremacy in certain problems, exemplified by Shor's algorithm for factoring and Grover's search algorithm [15]. In contrast to classical

computers, which adhere to deterministic laws of physics, microscopic quantum systems, when isolated from their environment, exhibit non-classical behaviors such as uncertainty, collapse, and entanglement [15,17].

Analogously to the bit in classical computing, the quantum bit or qubit is the basic unit of information processing used in quantum computing. Unlike classical bits, which can only exist in states of 0 or 1, qubits exploit the principles of superposition and entanglement. In superposition, a qubit can simultaneously occupy both states 0 and 1 until measured, providing a potential computational advantage over classical bits [15]. In quantum mechanics, the mathematical representation of qubits and their interactions is encapsulated within the framework of the Hilbert space. The state space of n qubits is described as a tensor product space, enabling the representation of complex quantum states and operations [15].

Quantum gates, the building blocks of quantum circuits, are unitary operators that perform transformations on qubits. These gates, akin to classical logic gates, perform operations on qubits, preserving the quantum information encoded within them. Through the sequential application of quantum gates, quantum circuits manipulate qubit states to perform specific computational tasks. Various quantum gates, including Pauli matrices and the Hadamard gate are utilized in circuits to perform operations on qubits, enabling the creation of entangled states and the implementation of quantum algorithms. These gates, in conjunction with control operations, form the basis of quantum circuits, which can be represented graphically to illustrate the flow of quantum information and operations [15].

2.2 Variational Quantum Circuits

Variational Quantum Circuits (VQCs) are trainable quantum circuits that are widely used as quantum neural networks for different tasks. VQCs are quantum algorithms that capture correlations in data using entangling properties [21]. In today's noisy intermediate-scale quantum computers (NISQ), which suffer from noise and qubit limitations, the VQC is the leading strategy due to their shallow depth [5].

In Fig. 1 the general schema of a VQC is shown. The first step is to encode the classical input x into a vector in the Hilbert space. This is accomplished by applying a unitary transformation $U_{\text{in}}(x)$ to the initial state, which is generally chosen as $|0\rangle^{\otimes n}$ [5]. After encoding the classical input, the state vector is passed through a set of quantum operations $U(\boldsymbol{\theta})$ depending on an optimizable parameter $\boldsymbol{\theta}$ [5]. Then, with U the encoding and trainable gates, the final state would be given as:

$$|\Psi\rangle = U\,|0\rangle^{\otimes n}\,, \tag{1}$$

Now, considering that the initialized state is $|0\rangle^{\otimes n} = (1, 0, \cdots 0)^T$, then the ith element of this state would be given as :

$$|\Psi_i\rangle = U_{ij}\delta_{j1} = U_{i1} \tag{2}$$

2.3 VQCs as Fourier series

A strategy presented in [23] and [4] is to have L layers, reuploading the encoded data each time. A layer consists of the encoding gates $S(x)$ and a trainable layer $W(\theta)$. The encoding $S(x)$ is composed of gates of the form $\exp\{ixH\}$, with H a Hamiltonian. This can always be decomposed as $H = V^\dagger \Sigma V$, with Σ a diagonal operator with the eigenvalues λ_i of H. The ij element of the transformation matrix after L layers can be written as:

$$U_{ij} = \sum_{j_1,\ldots,j_L=1}^{N} W^{(L+1)}_{ij_L} \exp(ix\lambda_{j_L}) W^{(L)}_{j_L j_{L-1}} \ldots W^{(2)}_{j_2 j_1} \exp(ix\lambda_{j_1}) W^{(1)}_{j_1 j}, \tag{3}$$

with $N = 2^n$ for n qubits. And finally, combining equations (2) and (3) the following result is obtained,

$$|\Psi\rangle_i = \sum_{j_1,\ldots,j_L=1}^{N} \exp(i(\lambda_{j_1} + \cdots + \lambda_{j_L})x) \times W^{(L+1)}_{ij_L} \ldots W^{(2)}_{j_2 j_1} W^{(1)}_{j_1 1} \tag{4}$$

In a VQC, a physical quantity, represented by the observable M, is measured as the quantum state passes through the circuit. This measurement yields the expected value, expressed as:

$$\langle M \rangle = \langle \Psi | \, M \, | \Psi \rangle \,, \tag{5}$$

Typically, the Pauli Z operator is used. The circuit is run multiple times (S repetitions), and the expected value is obtained by averaging measurements. This outcome is associated with labels by assigning probabilities, and a cost function is computed for parameter optimization [5,3,20].

Finally, combining the equation (4) and (5) and using the multi-index $\boldsymbol{j} = \{j_1, ..., j_L\} \in [N]^L$ and the sum of eigenvalues for $\boldsymbol{j}$ as $\boldsymbol{\Lambda_j} = \lambda_{j_1} + \cdots \lambda_{j_L}$, the result after the measurement can be seen as:

$$f(x) = \sum_{\boldsymbol{k},\boldsymbol{j} \in [N]^L} c_{\boldsymbol{k},\boldsymbol{j}} \exp\{i(\Lambda_{\boldsymbol{k}} - \Lambda_{\boldsymbol{j}})\}, \tag{6}$$

with,

$$c_{\boldsymbol{k},\boldsymbol{j}} = \sum_{i,i'} W^{*(1)}_{1 j_1} W^{*(2)}_{j_1 j_2} \ldots W^{*(L)}_{j_L j_{L-1}} W^{*(L+1)}_{j_L i} M_{ii'} W^{(L+1)}_{i' j_L} W^{(L)}_{j_L j_{L-1}} \ldots W^{(2)}_{j_2 j_1} W^{(1)}_{j_1 j} \tag{7}$$

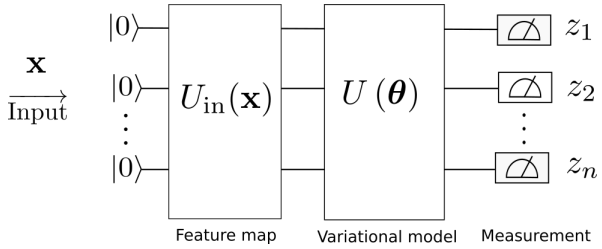


Fig. 1: Variational Quantum Circuit (VQC's). The classical input $\mathbf{x}$ is encoded into a quantum state $|\Psi_{\text{in}}(\mathbf{x})\rangle$ with the unitary transformation $U_{\text{in}}(\mathbf{x})$ applied to the initial quantum state $|0\rangle^{\otimes n}$. In the second step, a unitary transformation $U(\boldsymbol{\theta})$ with trainable parameters is applied. Finally, a measurement is made on the qubits.

Now, if the terms with the same frequency $\omega = \Lambda_k - \Lambda_j$ in the sum in (6) are grouped, the form of a real Fourier series is obtained [23] [4]:

$$f(x) = \sum_{\omega} c_\omega e^{i\omega x}. \tag{8}$$

In this last expression, the coefficients are the summation of all the $c_{\boldsymbol{k},\boldsymbol{j}}$ that contribute to the same frequency and as it can be seen in equation (7), they only depend on the trainable and measurement part of the circuit. On the other hand, the frequency spectrum is related to the encoding part. As it was proven in [23], the accessible frequency spectrum of a circuit can be linearly incremented by reuploading the encoding L times. Also, they showed how different architectures for the trainable part can produce different subsets of Fourier coefficients and set some coefficients to zero.

Although the capacity of QNNs needs to be further explored, several studies show that in certain cases they offer advantages in terms of the number of parameters and trainability [21,1,14].

3 1D Quantum convolution as multidimensional Fourier series

The 1D quantum convolution proposed in [18] takes the approach presented in [8], but with the difference that is adapted to one dimension, and the quantum layer is trainable. Instead of employing element-wise matrix multiplication as in the classical convolution, the 1D quantum convolution processes subsections of one-dimensional signals using a Variational Quantum Circuit (VQC) to generate a feature map. This VQC takes a sliding window of the input tensor, encoding the considered part into an initialized quantum state. A parameterized quantum circuit, represented by the unitary transformation $W(\boldsymbol{\theta})$ is applied. The trainable parameter $\boldsymbol{\theta}$ is determined during training. Subsequently, information is

decoded through measurements on all qubits, yielding real-number outputs (Eq. 5). Quantum measurements on each qubit result in individual output channels, with the number of output feature maps corresponding to the number of qubits. The schema of the 1D Quantum Convolution proposed in [18] is depicted in Fig. 3.

As the outputs of the 1D quantum convolution consist of real component vectors, this quantum layer can be effortlessly incorporated into diverse architectures, whether quantum or classical. In a study by [18], a hybrid approach is explored by integrating a 1D quantum convolution with a classical 1D convolution and linear layers. The model's performance is benchmarked against a purely classical counterpart, resulting in superior metrics for the hybrid configuration. In contrast, our focus in this work is to assess and compare the capabilities of different quantum architectures in achieving optimal outcomes. In this work, 1D quantum convolution is applied to the problem of time series forecasting: multiple points are encoded into the quantum circuit to predict the next one.

Since the problem treated in this work involves the encoding of multiple points, it is necessary to introduce the multidimensional version of the Fourier analysis presented in the previous section. In [4] an extension of the results in [23] is presented: they analyze the case when a multidimensional input is introduced in the quantum circuit and they propose several architectures that generate multidimensional truncated Fourier series given by the following equation,

$$f(\boldsymbol{x}) = \sum_{\omega_1,\ldots,\omega_M=-D}^{D} c_{\boldsymbol{\omega}} \exp\{i\boldsymbol{x}\cdot\boldsymbol{\omega}\}. \tag{9}$$

Here, $D = \max(\omega_1, ..., \omega_M)$ is the degree of the Fourier series, $\boldsymbol{x}$ is an M dimensional vector and the complex coefficients $c_{\boldsymbol{\omega}}$ have the property $c_\omega = c^*_{-\omega}$.

The time series that will be utilized in this work are unidimensional but the problem can be treated as multidimensional since several points are needed to predict the next point. Each time step can be seen as a dimension, it represents a feature, so encoding multiple points involves working with a multidimensional feature space in which each point contributes to a different dimension in this feature space.

The degrees of freedom ν of a Fourier series are the number of necessary independent variables to characterize a set of coefficients for a certain degree D and are given by the following expression:

$$\nu = (2D+1)^M. \tag{10}$$

As a minimum condition, the proposed architectures are required to have more or an equal number of trainable parameters than the degree of freedom to have a general enough circuit. In [4] several architectures are proven to output a

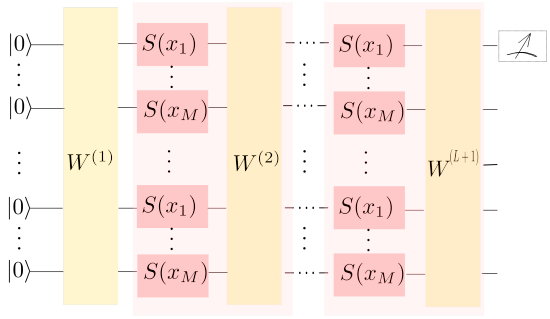


Fig. 2: Super parallel architecture proposed in [4] which outputs a multidimensional Fourier series. This general schema will be followed in this work.

multidimensional Fourier series but only the one called *super parallel ansatz* always satisfies the condition of having more free parameters than the degrees of freedom for any degree D. This architecture, illustrated in Figure 2, consists of stacking layers in depth and width directions. It requires $n = LM$ qubits and each layer can be represented as:

$$L(\boldsymbol{x}, \boldsymbol{\theta}) = \left(\bigotimes_{i=1}^{L} \left(\bigotimes_{m=1}^{M} S(x_m) \right) \right) W^{(l)}(\boldsymbol{\theta}). \tag{11}$$

The number of free parameters for this model is $N_p = (L+1)(2^{2ML} - 1)$, considering that, as it is proposed in [4], the unitary gates composing the trainable part have $N^2 - 1$ trainable parameters. This approach represents difficulties in the times at the training stage, as it will be discussed in the following section.

4 Expressibility

In [25] the expressibility is described as a circuit's ability to generate states representative of the Hilbert space. They propose to calculate it by comparing the distribution of the states generated by the circuit with the distribution of the Haar random states.

Given a dimension N, the unitary matrices of $N \times N$ constitute the unitary group $U(N)$. The Haar measure tells how to weight elements of $U(N)$ when performing operations with the unitary group [6]. An ensemble of Haar random states or t-design means that the first t moments of the Haar measure can be exactly captured: reproducing higher moments better approximates the full unitary group. Now, a frame potential is defined as:

$$\mathcal{F}^{(t)} = \int_{\theta} \int_{\phi} \| \langle \psi_\theta | \, | \psi_\phi \rangle \|^{2t} d\boldsymbol{\theta} d\boldsymbol{\phi}. \tag{12}$$

For an ensemble of Haar random states, $\mathcal{F}^{(t)} = \frac{t!(N-1)!}{(t+N-1)!}$ with $N = 2^n$ for n qubits. This value is a lower bound for the frame potentials: $\mathcal{F}^t \geq \mathcal{F}^{(t)}_{Haar}$. In [25] it is observed that potentials can be seen as the distribution of state overlaps: $P(F = \|\langle\psi_\theta|\,|\psi_\phi\rangle\|^2)$, with F the fidelity. The analytical form of the probability density function of fidelities is known for the ensemble of Haar random states: $P_{\text{Haar}}(F) = (N-1)(1-F)^{N-2}$. They propose to estimate the fidelity distribution by sampling pairs of states to obtain the probability distribution of fidelities with a histogram. With this, the expressibility is calculated with the Kullback-Leibler divergence between the estimated fidelity distribution and that of the Haar-distributed ensemble:

$$E = D_{KL}(P_c(F)|P_{\text{Haar}}(F)) \tag{13}$$

A lower KL divergence with respect to the Haar distribution corresponds to a more expressible circuit. The upper bound of expressibility is given by $(N-1)\ln(b)$, with b the number of bins in the histogram.

5 Barren plateaus

A circuit is said to exhibit a barren plateau if the gradient vanishes exponentially when adding more qubits, leading the optimizer to be trapped in a local minimum. The average of $\partial_k C$, with C the cost function over the set of trainable parameters $\boldsymbol{\theta}$ is zero, meaning that the cost gradients are not biased in any direction. In [9], the trainability of an ansatz is assessed with the Chebyshev inequality, which bounds the probability that the partial derivative deviates from its average of zero:

$$P(|\partial_k C| \geq \delta) \leq \frac{\text{Var}[\partial_k C]}{\delta^2}. \tag{14}$$

From this equation, it can be interpreted that when the variance is small, the gradient vanishes with high probability and those cases are untrainable. In [25], it is found that expressive ansatz exhibits barren plateaus.

6 Proposed model

In this work, the *super parallel ansatz* will be the general schema to follow, varying the part $W(\boldsymbol{\theta})$ to compare the performance both in metrics and in the subset of coefficients that each choice of $W(\boldsymbol{\theta})$ is capable to achieve. As mentioned above, utilizing gates of $N^2 - 1$ trainable parameters would imply large training times. For this reason, in this work the choice of $W(\boldsymbol{\theta})$ will be layers with only $n \times n$ trainable parameters (Fig 4(a)), which still fulfills the condition $N_p > \nu$. Also, those results will be compared against choices of $W(\boldsymbol{\theta})$ with only n (Fig 4(b),(d),(e)) and $3 \times n$ (Fig 4(c)) trainable parameters per layer, which do not fulfill the condition $N_p > \nu$.

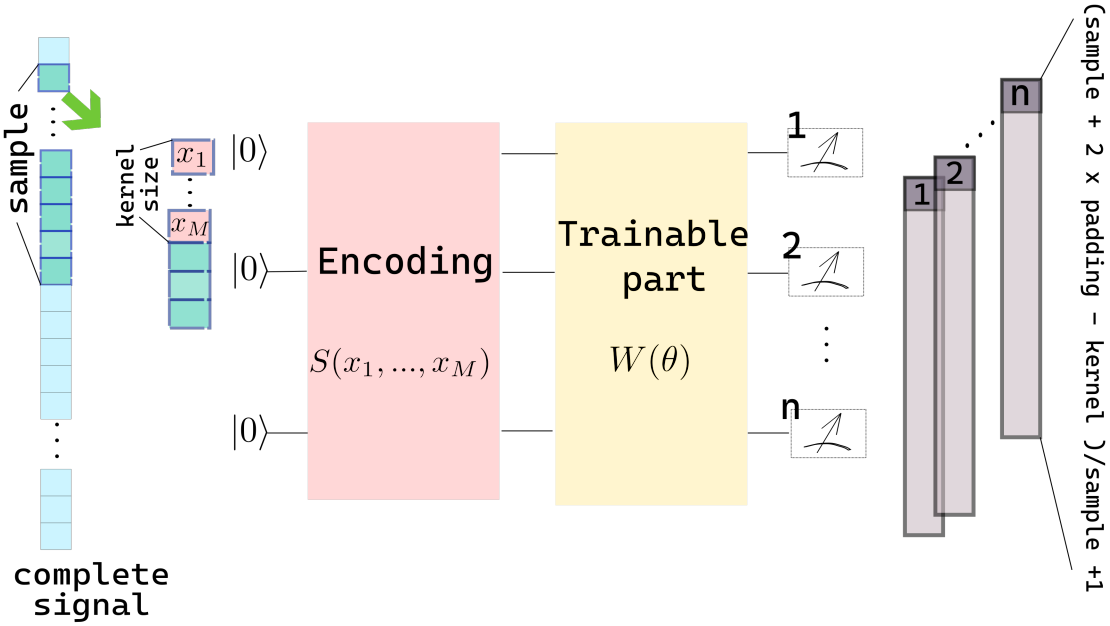


Fig. 3: 1D quantum convolution as introduced in [18]

The objective of this work is to compare the performance of different choices of $W(\boldsymbol{\theta})$, then the encoding part is the same for all the tests and is described in the following equation,

$$\bigotimes_{m=1}^{M} R_y(x_m). \tag{15}$$

This is repeated *vertically* L times in each of the L layers of the circuit, as illustrated in Fig 2.

A number c of points of the time series are taken to predict the next one and the 1D quantum convolutional circuit sweeps those points with a kernel of size k (with $k < s$). Then, the architecture of the circuit depicted in Fig 2 requires $k \times L$ qubits. This is an advantage of utilizing quantum convolution instead of the quantum version of a multilayer perceptron in which $s \times L$ qubits would be required if the *super parallel* architecture was followed.

The sliding window size for the experiments in this work is $k = 2$ and c varies for each dataset. This effectively allows the circuit to capture the dependencies between adjacent points. The *super parallel* architecture will have 2 layers, which is its simplest case. With 2 layers and $k = 2$, 4 qubits are required. For this architecture, the degree of the output Fourier series is expected to be $D = 4$, but depending on the particular choice of $W(\theta)$ some of the coefficients might be set to zero, limiting the expressivity of the model. The observable R_z is measured in those 4 qubits at the end of the circuit, and the expected value (a real number) of each qubit is passed to the classical part. The output features of the 1D quantum convolution is a matrix with dimensions (n, o), with $o = (c + 2 \times p - k)/s + 1$. Here, p is the padding and s is the stride. In this case, the sample chain of size c is padded at both edges with a zero. The chain is swept with a stride $s = 1$. The classical part consists of a ReLU activation function, a max pooling over the dimension o of the output of the quantum convolution, and a linear layer that maps from n elements to finally output the predicted point of the time series.

The relevance of reuploading data is assessed by comparing the RMSE, MAE and MAPE metrics. Also, a comparison between the parallel and super-parallel architecture is performed. Finally, the performances of ansatz in Figure 4 are calculated. Ansatz (c) is known as Strongly Entangling Layers, is a template from *Pennylane* inspired by the work presented in [22]. Ansatz (d) is the temple known as *Basic Entangler*, also from *Pennylane*, with a similar structure as Strongly Entangling but with a rotation only in one direction. Ansatz (e) is generated with the template *Random Layers* of *Pennylane* and seed=1234, which randomly distributes two-qubit gates and rotations in the circuit. The number of random rotations is derived from the second dimension of weights and the number of CNOT gates is $\frac{1}{3}$ of the number of rotations.

To compare the expressivity of each of the architectures depicted in Fig 4, the Fourier coefficients will be computed. Also, the expressibility as described in [25] and the variance of the gradient of the cost function as described in [9] are calculated. To calculate the expressibility, the calculations were generated by sampling 5000 states and using a bin size of 75, following the parameters suggested in [25]. The circuit is initialized with different sets of random weights and the coefficients are computed from the output of the circuit using a discrete Fourier transform. Those coefficients can be plotted, allowing to easily evaluate the richness of the accessible coefficients of each architecture. The analytical expression of the part of the circuit corresponding to the coefficients is highly non-linear and too complex, then a graphical analysis is more suitable to provide an insight into the differences between each choice of the variational part.

The computations were executed using *Pennylane* on the simulator device *default.qubit*, employing the *torch* interface and the *backprop* differentiation method. The calculations were performed on a Ryzen 7 5700X processor.

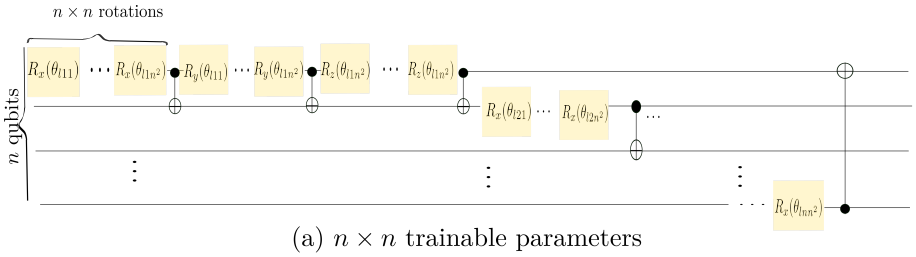
6.1 Data

The datasets that will be analyzed in this work are the third Legendre polynomial with random seeded noise, the Mackey glass time series, and the exchange rate between USD and EUR.

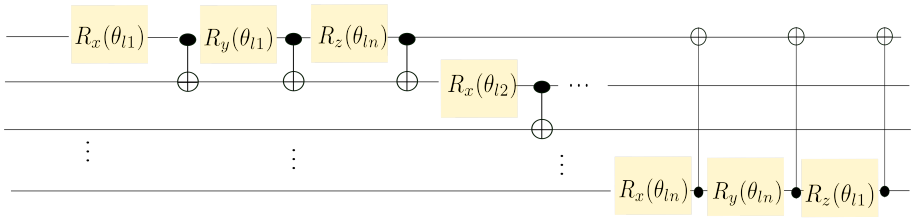
The third Legendre polynomial is given as $P_3(x) = \frac{1}{2}(3x^2 - 1)$. The dataset consists of points given by this equation, adding at each point random noise. The samples that will be swept by the 1D convolution kernel consist of 5 points. 750 points are utilized for training and 250 for testing.

The Mackey-Glass time series data is derived from a differential equation characterized by parameters α, β, and γ, with τ representing the time delay. Numerical solutions are obtained using the Runge-Kutta method, initialized with $x(0) = 1.2$ and a step size of 0.1. The specific values for α, β, and γ are 0.2, 0.1, and 10, respectively. Our model is constructed using 1000 simulation data points, defined as $[x(t-18), x(t-12), x(t-6), x(t); x(t+6)]$ for t ranging from 19 to 1018. The first 500 points serve as training data, and the remaining points are allocated for testing. The model input comprises the vector x with components $x(t-18)$, $x(t-12)$, $x(t-6)$, and $x(t)$, while the output variable is the last component $x(t+6)$.

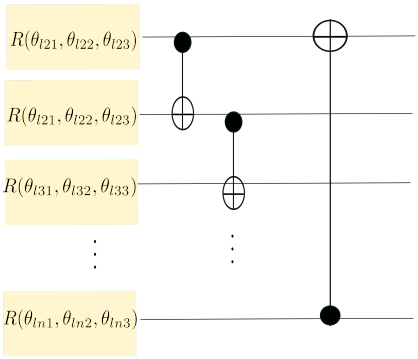
The exchange rate data between USD and EUR, sourced from [26], spans from January 1, 2020, to July 8, 2021, with daily observations. Constructing our model involves 376 simulation data points, given by the sequence $[x(t-4), x(t-3), x(t-2), x(t-1), x(t); x(t+1)]$ for t ranging from 5 to 380. The first 300 data points are designated for training, and the remaining points are reserved for testing. The expressibility measured as proposed in [25] is calculated for both cases



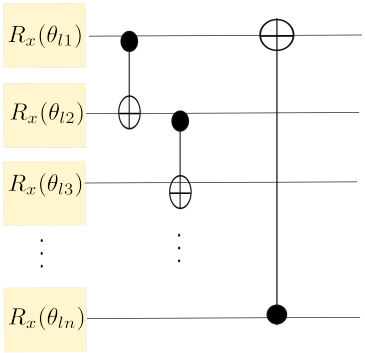
(a) $n \times n$ trainable parameters



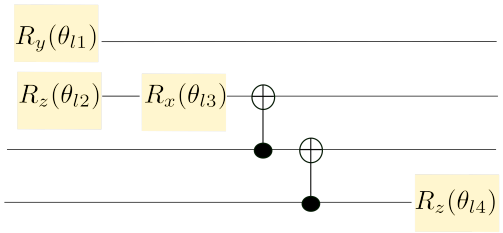
(b) n trainable parameters



(c) $3 \times n$ trainable parameters



(d) n trainable parameters



(e) n trainable parameters

Fig. 4: Ansatz for the trainable part $W(\theta)$

7 Results and Discussion

7.1 Number of necessary trainable parameters

The sets of accessible coefficients for each of the ansatz in Fig 4 are depicted in Figure 5. Those results were obtained with 100 different sets of randomly initialized weights. It can be observed that even when the architectures (b), (c), and (d) do not fulfill the condition $N_p > \nu$, they are capable of producing non-zero coefficients. Moreover, it can be noted that the architectures (c) and (d) are capable of producing a richer set of coefficients than (a), which fulfills the condition. The expected degree for all the architectures is 4, then according to Equation 10, at least 81 free parameters would be needed to have a general enough circuit. It is noticeable that (b) and (d) with as few as 12 trainable parameters and (c) with 36 trainable parameters are capable of producing more non-zero coefficients than expected. (b) is capable of producing coefficients corresponding to a Fourier series of degree $D = 2$, which would need 25 free parameters. (c) and (d) have non-zero coefficients corresponding to a degree $D = 3$, for which 49 free parameters would be needed.

In practical terms, achieving high-degree Fourier functions with fewer trainable parameters is highly desirable due to the difference in training times. For the ansatz (b), (c), and (d) of Figure 4, with 4 qubits and 2 layers, training took on average 1950 s. On the other hand, ansatz (a) took on average 49500 s to complete 30 epochs. This represents a difference of 3000 %. Also, the circuits with fewer parameters were capable of achieving comparable metrics, to illustrate this, the results for the Mackey glass dataset are presented in Table 1. The other datasets exhibited similar behavior.

Ansatz	Trainable parameters	RMSE	MAE	MAPE	Training Time (s)
(a)	192	0.0810	0.0660	0.0759	43740
Strongly Entangling	36	0.07681	0.06351	0.07015	2028
Basic Entangler	12	0.0860	0.0725	0.0854	1610
Custom Layers	12	0.0952	0.0772	0.0967	3567
Random Layers	12	0.1170	0.0980	0.1196	1484

Table 1: Comparison of results obtained with the ansatz of Figure 4 for the Mackey glass dataset.

The degree D of the Fourier series for the parallel architecture is equal to the number of layers L and for the super-parallel architecture is given by the square of the layers L. When more qubits and layers are added, ansatz (b), (c), and (d) are capable of producing more non-zero terms than expected by their respective

number of trainable parameters. This can be appreciated in Table 2: in column 6, the actual trainable parameters of each case are presented and in column 7 the parameters needed to fulfill the condition $N_P > \nu$, being the values in column 7 significantly larger than in column 6.

Ansatz	Qubits	Layers	Expected degree	Obtained degree	Trainable parameters	Parameters needed for obtained D
Strongly Entangling	6	3	9	8	72	289
	8	4	16	11	120	529
Basic Entangler	6	3	9	8	24	289
	8	4	16	10	40	441
Custom Layers	6	3	9	5	24	121
	8	4	16	8	40	289
Random Layers	6	3	9	2	24	25
	8	4	16	2	40	25

Table 2: Degree obtained for 6 and 8 qubits with 3 and 4 layers, respectively. In the last column, the parameters that in theory would be needed to fulfill the condition $N_p > \nu$.

Having a small number of free parameters producing more non-zero degree coefficients than expected is due to the highly coupled and non-linear form of the coefficients equation; the relationship between the parameters and the Fourier coefficients is highly non-linear, allowing the model to capture complex dependencies with a small number of parameters.

7.2 Relevance of reuploading

A comparison between the results of the metrics with and without reuploading is presented. In this case, 2 circuits are compared. For the non-reuploading case, an initial trainable layer $W^{(1)}$, the encoding of the information of the 2 points kernel into 2 qubits, followed by 4 layers of the trainable ansatz. This is compared against the parallel architecture with 2 qubits and 4 layers. The parallel architecture is similar to the super-parallel, but without reuploading the data vertically: the data is only repeated in a series of layers. The values of the expressibility (calculated as proposed in[25]) for each of the trainable layers in Figure 4 are calculated for both cases and depicted in Table 3. It can be observed that both cases present comparable values, but an analysis of the metrics in the testing stage for each data set shows that the reuploading strategy is superior. This behavior can be explained if the accessible coefficients for each case are analyzed. In Figure 6 a comparison between the accessible coefficients for the parallel and the case without reuploading is presented. It can be observed a clear difference between both architectures, showing that when no reuploading of data

is performed, the circuit has significantly less accessible frequencies.

Ansatz	Expressibility		Average % improvement by using reuploading		
	Parallel	Non-reuploading	RMSE	MAE	MAPE
Strongly Entangling	0.0071	0.0082	45.88	47.19	23.11
Basic Entangler	0.0210	0.0375	56.24	57.94	33.59
Custom Layers	0.0090	0.0100	23.47	28.60	33.59
Random Layers	0.8729	0.8784	34.59	36.25	7.29

Table 3: Expressibility for the parallel architecture vs non-reuploading architecture and average taken over the three datasets of improvement (diminution of RMSE, MAE and MAPE) by using reuploading

7.3 Parallel vs Super parallel architectures

The performance of the parallel and super-parallel architecture presented in [4] is compared. A super-parallel architecture with 4 qubits and a kernel of 2 is compared against a parallel architecture with 2 qubits, a kernel of 2, and 4 layers. In this way, the data is re-uploaded the same number of times and the circuits are expected to output Fourier series of the same degree $d = 4$. In general, as can be observed in Table 4, the super-parallel architecture produced better results overall. In this case, no major difference between the accessible coefficients to both architectures is presented, but as can be noted in Table 4, the values for the expressibility are better for the super-parallel case. Even when training times are shorter for the parallel case, the difference is not significant enough.

Ansatz	Expressibility		Average % improvement by using Super Parallel			
	Parallel	Super Parallel	RMSE	MAE	MAPE	Training time % difference
Strongly Entangling	0.0071	0.0033	26.11	26.64	24.81	16.18
Basic Entangler	0.0210	0.0053	5.991	7.165	3.51	15.63
Custom Layers	0.0090	0.0054	5.46	3.07	-0.304	13.71
Random Layers	0.8729	2.764	27.12	24.85	37.62	14.57

Table 4: Values for the expressibility of the parallel and super parallel architecture for each different ansatz, along with the corresponding percentage improvements in metrics when employing the super-parallel architecture compared to the parallel architecture.

7.4 Comparison between ansatz

The ansatz (b)-(e) in Figure 4 are tested for 4, 6, and 8 qubits with 2, 3, and 4 layers respectively. In Table 5 the values of the degree of the obtained Fourier function, the expressibility, and the variance of the derivative of the cost function are presented. It can be observed a relation between the accessible coefficients and the expressibility, both parameters are complementary. The ansatz (e) has only 2 accessible coefficients for the 3 tested cases and also presents a high value of the Kulbak-Leibler divergence, significantly different from (b)-(d). As it can be noted from Figure 4, the difference between this ansatz and the other 3 is the number of entangling CNOT gates; while the other cases have all their qubits entangled, in (e) only $\frac{1}{3}$ of the qubits are entangled. The results for (b)-(d) are comparable, but some differences can be noted. The Basic Entangler and the Strongly Entangling have a similar structure: each qubit is rotated first and after that, all qubits are entangled. This similarity results in the closeness of the 3 analyzed values; those are also slightly different from Custom Layers (which has a different. structure), especially in the manner that the values evolve when adding more qubits. This impact of the structure of an ansatz can also be noted in Figure 5: (a) and (d) have the same accessible coefficients despite the difference of trainable parameters (192 and 12, respectively) and the same happens for (b) and (c). The results for (b) are superior to (c). This is easily explainable since (b) performs a rotation in the 3 directions, being able to cover the Hilbert space in a more complete form. Also, (b) has 3 times more trainable parameters than (c). The expressibility and the accessible coefficients for (b) grow when adding more qubits, but this also implies a lower value for the variance of the derivative of the cost function, which might result in a flat cost landscape. Ansatz (d) presents lower values for expressibility and has less accessible coefficients than (b) and (c), but in the other hand the value of the variance evolves slowly, implying better trainability.

Assessing the impact of the obtained degree values, expressiveness, and variance with real data is essential. For this purpose, the datasets described in subsection 3.1 are utilized with the hybrid model described in section 3.

The Legendre polynomial dataset exhibited better results when utilizing 8 qubits for the 4 ansatz. With 4 qubits, the Strongly Entangling ansatz achieved the best results, followed by the Custom Layers. For 6 qubits, the best result corresponds to the Custom Layers in MAE and RMSE, followed by Basic Layers and Strongly Entangling. Only in MAPE, the results of Strongly Entangling are superior. For 8 qubits, as shown in Figure 7, the best result is given by Basic Layers, closely followed by Strongly Entangling.

For the Mackey Glass dataset, in the 4 tested ansatz, the best results were obtained by utilizing 8 qubits, followed by 6 qubits and 4 qubits in the last place, as expected. Analyzing the case with 8 qubits, the architecture with the best results is Strongly Entangling, followed by Basic Entangler which presented sim-

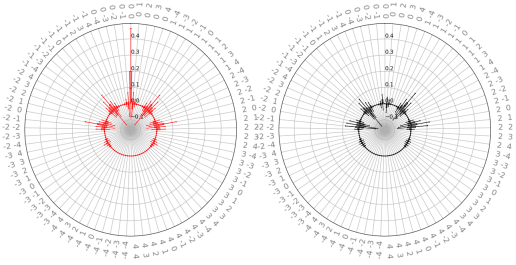
Ansatz	Qubits	Layers	Obtained degree	Expressibility	Variance of derivative
Strongly Entangling	4	2	3	0.0033	0.021
	6	3	8	0.0013	0.0063
	8	4	11	0.00011	0.0016
Basic Entangler	4	2	3	0.0053	0.0434
	6	3	8	0.0008	0.0132
	8	4	10	0.0004	0.0032
Custom Layers	4	2	2	0.0054	0.0454
	6	3	5	0.0075	0.0223
	8	4	8	0.0032	0.0112
Random Layers	4	2	2	2.764	0.1270
	6	3	2	5.604	0.0860
	8	4	2	1.737	0.0605

Table 5: Degree, expressibility, and variance of the derivative of the cost function for the ansatz (b)-(e) in Figure 4. The cases with 4, 6, and 8 qubits are analyzed.

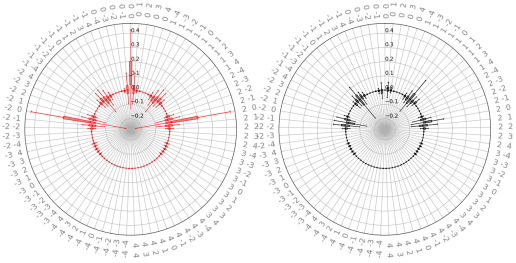
ilar results to Custom Layers. This is depicted in Figure 8.

In the Euro dataset, the values of the metrics seem to be more sensitive to the results of Table 5. To better appreciate the difference in results by changing the number of qubits, Figure 9 is plotted differently than the results for the Mackey Glass and the Legendre polynomial. The Random Layers and the Custom Layers exhibited a lower expressibility for 6 qubits, which is reflected in the metrics, as can be observed in Figure 9. In the case of Strongly Entangling and Basic Entangler, the values of the expressibility increase with the number of qubits, but this comes with a decrease in the divergence of the gradient of the cost function. This impacts the obtained metrics, the results for 8 qubits are worse than for 6 qubits in both cases. The Strongly Entangling configuration exhibits a lower gradient divergence than the Basic Entangler. This leads to a more pronounced difference between 6 and 8 qubits in the case of Strongly Entangling as opposed to Basic Entangler. This behavior can be seen in Figure9.

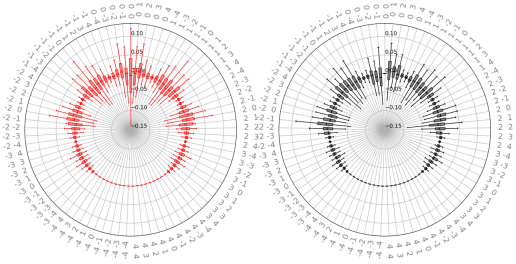
Across the three datasets, the Random Layers ansatz yielded less favorable outcomes, as anticipated due to its limited expressibility and absence of non-zero Fourier coefficients. In the context of the Legendre polynomial and Mackey Glass datasets, optimal results generally manifested with the more expressive ansatz options—specifically, the Strongly Entangler and the Basic Entangler with 8 qubits—despite their relatively lower variance in the gradient of the cost function. In contrast, the Euro dataset proved to be more challenging to train using the most expressive ansatz. One hypothesis is that the Legendre and Mackey Glass datasets exhibit greater resilience to a flatter cost landscape, potentially attributed to having the double training samples compared to the Euro dataset.



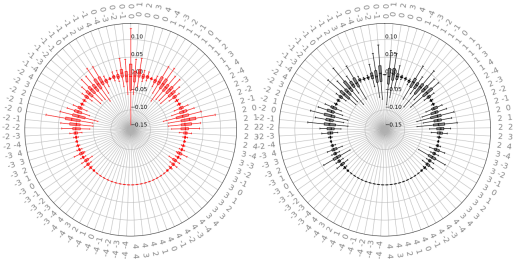
(a) $n \times n = 192$ trainable parameters



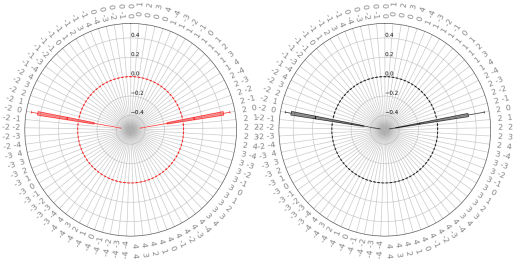
(b) $n \times L = 12$ trainable parameters



(c) $3 \times n \times L = 36$ trainable parameters



(d) $n \times L = 12$ trainable parameters



(e) $n \times L = 12$ trainable parameters

Fig. 5: Accessible coefficients for the architectures of Fig 4

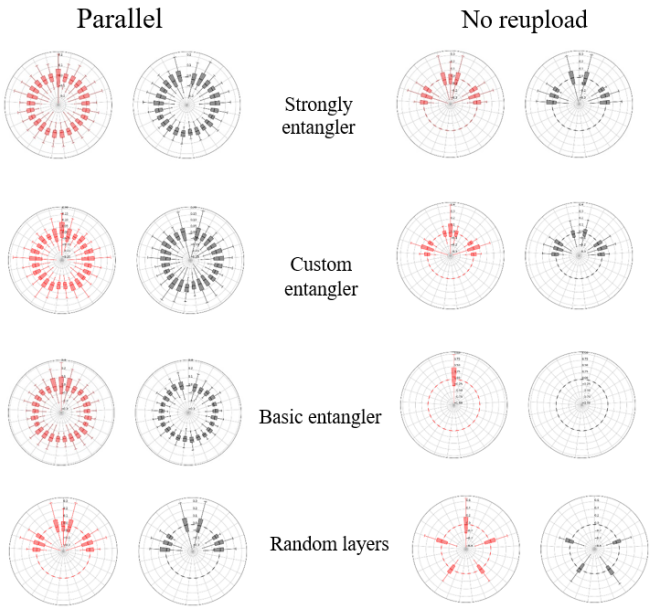


Fig. 6: Comparison of the accessible coefficients for the parallel and the case without reuploading. The ansatz are depicted in Figure 4.

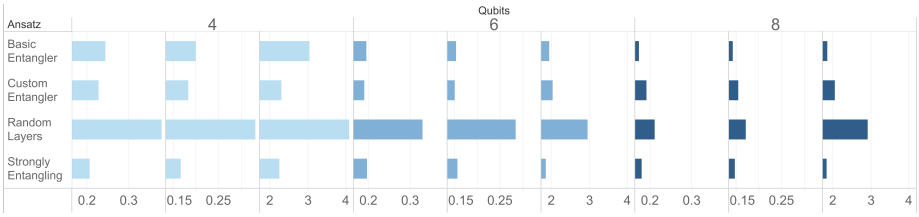


Fig. 7: Comparison between ansatz for the Legendre polynomial dataset with 8 qubits and 4 layers. The obtained results with this number of qubits were superior to 4 and 6 qubits.

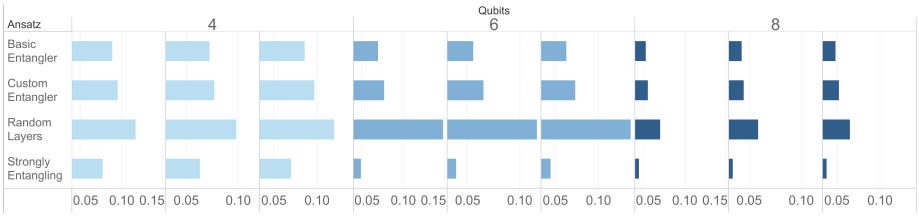


Fig. 8: Comparison between ansatz for the Mackey glass dataset.

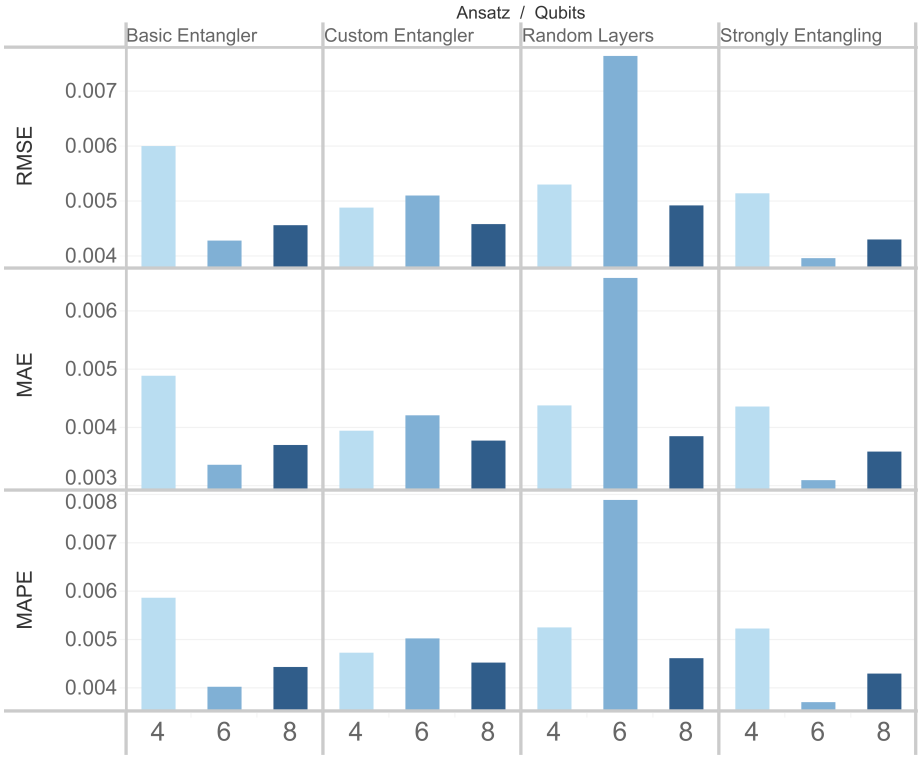


Fig. 9: Comparison between 4,6 and 8 qubits for the Euro dataset.

8 Conclusions

In this study, we leverage the theoretical insights from [23], [4], [25], and [9] to evaluate the performance of four different ansatz, considering practicality in training with real datasets. While numerous ansatz and qubit combinations exist, exploring all permutations is beyond the scope of this work, and results may vary with different datasets. Nevertheless, key conclusions can be drawn from our findings.

Contrary to the condition emphasized in [4] that $N_p > \nu$ is necessary, our results suggest that a limited number of trainable parameters can yield Fourier functions of higher degrees, underscoring the remarkable expressive power of quantum circuits. This observation is particularly pertinent in terms of training times and offers insight into why previous works achieved good results with few parameters might be successful. Further work needs to be done to provide more information about why few trainable parameters are capable of producing higher degree Fourier series than expected.

By analyzing the problem of quantum 1D convolution in the context of Fourier transforms, it was possible to design the architecture in a form in which data is reuploaded, which lead to significantly improved results. This improvement can be infered from the theory presented in [23] and [4], where they also pointed time series as a natural application to their work. Employing a super-parallel structure proves more effective than reuploading the data an equivalent number of times in a parallel structure. The significance of data reuploading might not have been evident solely from an expressibility analysis: it only becomes apparent when considering accessible coefficients, and the superiority of the super-parallel approach cannot be inferred from the coefficients but is noted when analyzing the expressibility. Then, it can be said that both parameters are complementary, emphasizing the importance of a comprehensive analysis when seeking an optimal ansatz.

Regarding specific ansatz performances, Strongly Entangler, Custom Entangler, and Basic Entangler consistently yield favorable results, with a tendency for improved metrics as the number of qubits increases. In general, the Strongly Entangling architecture, being the most expressive, also demonstrates superior performance. The expressibility of Strongly Entangling also comes with the price of having a flat optimization landscape when increasing the number of qubits. When analyzing the testing metrics, it can be seen that this only affected the Euro dataset, which had fewer training points. In the cases with more training points, the expressibility of the ansatz was more important. These findings contribute valuable insights for selecting effective ansatz configurations in quantum machine learning applications.

References

1. Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., Woerner, S.: The power of quantum neural networks. Nature Computational Science **1**(6), 403–409 (2021)
2. Alejandra, R.R.M., Andres, M.V., Mauricio, L.R.J.: Time series forecasting with quantum machine learning arquitectures. In: Obdulia Pichardo Lagunas, Juan Martínez-Miranda, B.M.S. (ed.) Advances in Computational Intelligence. pp. "66–82" (2022)
3. Bergholm, V., Izaac, J., Schuld, M., Gogolin, C.: Pennylane: Automatic differentiation of hybrid quantum-classical computations (2022)
4. Casas, B., Cervera-Lierta, A.: Multidimensional fourier series with quantum circuits. Phys. Rev. A **107**, 062612 (Jun 2023). https://doi.org/10.1103/PhysRevA.107.062612, https://link.aps.org/doi/10.1103/PhysRevA.107.062612
5. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., et al.: Variational quantum algorithms. Nature Reviews Physics **3**(9), 625–644 (2021)
6. Di Matteo, O.: Understanding the haar measure (Mar 2021), https://pennylane.ai/qml/demos/tutorial_haar_measure
7. Feynman, R.P.: Simulating physics with computers. In: Feynman and computation, pp. 133–153. CRC Press (2018)
8. Henderson, M., Shakya, S., Pradhan, S., Cook, T.: Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Machine Intelligence **2**(1), 2 (2020)
9. Holmes, Z., Sharma, K., Cerezo, M., Coles, P.J.: Connecting ansatz expressibility to gradient magnitudes and barren plateaus. PRX Quantum **3**, 010313 (Jan 2022). https://doi.org/10.1103/PRXQuantum.3.010313, https://link.aps.org/doi/10.1103/PRXQuantum.3.010313
10. Hong, Z., Wang, J., Qu, X., Zhu, X., Liu, J., Xiao, J.: Quantum convolutional neural network on protein distance prediction. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2021)
11. Houssein, E.H., Abohashima, Z., Elhoseny, M., Mohamed, W.M.: Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images. Journal of Computational Design and Engineering **9**(2), 343–363 (02 2022). https://doi.org/10.1093/jcde/qwac003, https://doi.org/10.1093/jcde/qwac003
12. Hur, T., Kim, L., Park, D.K.: Quantum convolutional neural network for classical data classification. Quantum Machine Intelligence **4**(1) (feb 2022). https://doi.org/10.1007/s42484-021-00061-x
13. Mari, A., Bromley, T.R., Izaac, J., Schuld, M., Killoran, N.: Transfer learning in hybrid classical-quantum neural networks. Quantum **4**, 340 (Oct 2020). https://doi.org/10.22331/q-2020-10-09-340, https://doi.org/10.22331/q-2020-10-09-340
14. Mitarai, K., Negoro, M., Kitagawa, M., Fujii, K.: Quantum circuit learning. Phys. Rev. A **98**, 032309 (Sep 2018). https://doi.org/10.1103/PhysRevA.98.032309, https://link.aps.org/doi/10.1103/PhysRevA.98.032309
15. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
16. Park, G., Huh, J., Park, D.K.: Variational quantum one-class classifier. Machine Learning: Science and Technology **4**(1), 015006 (jan 2023). https://doi.org/10.1088/2632-2153/acafd5, https://dx.doi.org/10.1088/2632-2153/acafd5

17. Preskill, J.: Quantum computing 40 years later. Nature Reviews Physics **4**(1) (jan 2023). https://doi.org/https://doi.org/10.1038/s42254-021-00410-6
18. Rivera-Ruiz, M.A., Juárez-Osorio, S.L., Mendez-Vazquez, A., López-Romero, J.M., Rodriguez-Tello, E.: 1d quantum convolutional neural network for time series forecasting and classification. In: Calvo, H., Martínez-Villaseñor, L., Ponce, H. (eds.) Advances in Computational Intelligence. pp. 17–35. Springer Nature Switzerland, Cham (2024)
19. Sameer, M., Gupta, B.: A novel hybrid classical-quantum network to detect epileptic seizures. medRxiv pp. 2022–05 (2022)
20. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., Killoran, N.: Evaluating analytic gradients on quantum hardware. Physical Review A **99**(3) (mar 2019). https://doi.org/10.1103/physreva.99.032331, https://doi.org/10.11032Fphysreva.99.032331
21. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. Phys. Rev. A **101**, 032308 (Mar 2020). https://doi.org/10.1103/PhysRevA.101.032308, https://link.aps.org/doi/10.1103/PhysRevA.101.032308
22. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. Physical Review A **101**(3) (Mar 2020). https://doi.org/10.1103/physreva.101.032308, http://dx.doi.org/10.1103/PhysRevA.101.032308
23. Schuld, M., Sweke, R., Meyer, J.J.: Effect of data encoding on the expressive power of variational quantum-machine-learning models. Physical Review A **103**(3) (mar 2021). https://doi.org/10.1103/physreva.103.032430, https://doi.org/10.11032Fphysreva.103.032430
24. Shahwar, T., Zafar, J., Almogren, A., Zafar, H., Rehman, A., Shafiq, M., Hamam, H.: Automated detection of alzheimer's via hybrid classical quantum neural networks. Electronics **11**, 721 (02 2022). https://doi.org/10.3390/electronics11050721
25. Sim, S., Johnson, P.D., Aspuru-Guzik, A.: Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies **2**(12), 1900070 (2019). https://doi.org/https://doi.org/10.1002/qute.201900070, https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.201900070
26. Werner: Pacific exchange rate service. http://fx.sauder.ubc.ca/data.html (2023), accessed on 2023-01-20
27. Yang, Y.F., Sun, M.: Semiconductor defect detection by hybrid classical-quantum deep learning. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (jun 2022). https://doi.org/10.1109/cvpr52688.2022.00236, https://doi.org/10.11092Fcvpr52688.2022.00236