An exponentially-growing family of universal quantum circuits

Machine learning

An exponentially-growing family of universal quantum circuits

Mohammad Kordzanganeh, Pavel Sekatski, Leonid Fedichkin, and Alexey Melnikov

Terra Quantum AG

Kornhausstrasse 25, 9000 St. Gallen, Switzerland

Quantum machine learning has become an area of growing interest but has certain theoretical and hardware-specific limitations. Notably, the problem of vanishing gradients, or barren plateaus, renders the training impossible for circuits with high qubit counts, imposing a limit on the number of qubits that data scientists can use for solving problems. Independently, angle-embedded supervised quantum neural networks were shown to produce truncated Fourier series with a degree directly dependent on two factors: the depth of the encoding, and the number of parallel qubits the encoding is applied to. The degree of the Fourier series limits the model expressivity. This work introduces two new architectures whose Fourier degrees grow exponentially: the sequential and parallel exponential quantum machine learning architectures. This is done by efficiently using the available Hilbert space when encoding, increasing the expressivity of the quantum encoding. Therefore, the exponential growth allows staying at the low-qubit limit to create highly expressive circuits avoiding barren plateaus. Practically, parallel exponential architecture was shown to outperform the existing linear architectures by reducing their final mean square error value by up to 44.7% in a one-dimensional test problem. Furthermore, the feasibility of this technique was also shown on a trapped ion quantum processing unit.

I.   INTRODUCTION

The successes of quantum computing in the past decade have laid the foundations for the interdisciplinary field of quantum machine learning (QML), where parameterized quantum circuits (PQC) are used as machine learning models to extract features from the training data. It was argued [1] that such quantum neural networks (QNN) could have higher trainability, capacity, and generalization bound than their classical counterparts. Practically, hybrid quantum neural networks (HQNN) have shown promise in small-scale benchmarking tasks [2, 3] and larger-scale industrial tasks [4, 5]. Nevertheless, the utility, practicality, and scalability of pure QNNs are still unclear. Furthermore, [6] provided a thorough overview of this field, showing that while classical machine learning is solving large real-world problems, QNNs are mostly tried on synthetic, clean datasets and show no real-world immediate advantages in its current state[1]. It also suggested that the QNN research focus ought to be shifted from seeking quantum advantage to new research questions, such as finding a new, advantageous quantum neuron. This work explores a new quantum neuron that argues for moving beyond using the Pauli-, single-qubit gates for encoding data, and instead employing higher-dimensional unitaries through gate decomposition.

Since quantum gates are represented by elements of compact groups, Fourier analysis is a natural tool to analyse QNNs. [10] showed that certain quantum encodings can create an asymptotically universal Fourier estimator. A universal Fourier estimator is a model that can fit a trigonometric series on any given function and as the number of terms in the series approaches infinity, the fit becomes an asymptotically perfect estimate. This estimator can initially infer the coarse correlations in the supplied data, and by increasing the number of Fourier terms, it can incrementally judge the more granular properties of the dataset. This provides an adjustable, highly-expressive machine-learning model.

In [10], the authors showed that a QNN can operate as such in two ways: a *sequential* single-qubit architecture with n repetitions of the same encoding could yield n Fourier bases, which could also arise from an n-qubit architecture with the encoding gates applied to all qubits in *parallel*.

The sequential architecture is widely shown to be an efficient Fourier estimator, demonstrated both theoretically [10] and empirically [11, 12]. However, sequential circuits are often deep [12], and assuming that near-term quantum computers will have a non-negligible noise associated with each gate, these circuits can experience noise-induced barren plateaus [13]. Barren plateaus are a phenomenon observed in optimization problems where the gradients of the model vanish, rendering them impossible to train.

---

[1] This is true for pure quantum models trained on classical datasets. There are numerous successes in applying quantum models to quantum-native problems [7–9].

More importantly, a single qubit can be simulated efficiently in a classical setting, so this architecture brings no quantum advantage.

In contrast, the parallel setting offers the exponential space advantage of quantum computing but poses two challenges for large numbers of qubits:

- An exponentially growing parameter count with the number of qubits, required to span the entire group space $\mathcal{SU}(2^n)$ [14], which could also lead to noise-induced barren plateaus. Spanning the full space is especially important for a priori problems, where the best model lies somewhere in the Hilbert space, but the machine learning scientist has no prior knowledge of how to parameterize a circuit to reach this point. In addition, gradient calculations in QNNs – as of this publication – use the parameter-shift rule discovered and presented in [15]. This method requires two evaluations of the QNN to find the derivative of the circuit with respect to each of the trainable parameters. An exponentially growing number of trainable parameters translates to an exponentially increasing amount of resources required for gradient computation. In mitigation, [16, 17] showed that a polynomially-growing number of parameters could generate a similar result based on the quantum t-design limits [18].

- Strongly-parameterised QNNs have a vanishing variance of gradient which decreases exponentially with the number of qubits [19]. This means that for a large number of qubits if one were to initialize her QNN randomly, they will encounter a barren plateau. This happens because the expectation value of the derivative of the loss function with respect to each variable for any well-parameterized quantum circuit is zero, and its variance decreases exponentially with the number of qubits. Mitigation methods have been suggested in [20, 21], and most notably in [22] it was shown that by relaxing the well-parameterized constrain to only include a logarithmically growing circuit depth with the number of qubits in the system and use local measurements, the circuit is guaranteed to evade barren plateaus. [23] developed a platform based on ZX-calculus[2] to explore which QNN architectures are affected by the barren plateau phenomenon and found that strongly-entangling, hardware efficient circuits suffer from them. In contrast with the previously-mentioned cases of barren plateaus, the latter is not noise-induced and thus this is a problem that needs to be addressed even in the fault-tolerant future of quantum computing.

Therefore, the practicing QML scientist is limited in choosing her QNN architectures for general data science problems: they need to be shallow or employ only a few qubits.

This contribution suggests a modification to the encoding strategies in [10] to increase the growth of the Fourier bases in a QNN from linear in the number of qubits/number of repetitions, to exponential. The proposed encoding is constructed by decomposing large unitary generators into local Pauli-Z rotations. This improves the expressivity of the QNNs without requiring additional qubits or encoding repetitions. The increased expressivity is a product of eliminating the encoding degeneracies of the quantum kernel, making efficient use of the available Hilbert space by assigning a unique wave-vector to each of its dimensions. However, such encodings could introduce a greater risk of limiting the model's Fourier accessibility[3].

Sec. II provides a review of how angle-embedded QNNs approximate their input distributions by fitting to them a truncated Fourier. Specifically, Sec. II reviews the linear encoding architectures and how their number of Fourier bases grow linearly with the number of repetitions – sequential linear in Sec. II A – as well as the number of qubits – parallel linear in Sec. II B. Then, Sec. III introduces the same two architectures but with a slight modification to represent an exponentially-growing number of Fourier bases. To use these architectures in practice, Sec. III C showcases a comparison in the training performance of these architectures and shows that the parallel exponential has a superior training performance to the other architectures on a synthetic, one-dimensional dataset. Finally, Sec. III D critically evaluates the work and suggests areas for future investigation.

---

[2] A graphical language, based on tensor networks, to analyze quantum circuits[24].

[3] Each of these bases has a Fourier amplitude as well as a phase angle, both of which need to be altered to fit a model on the training data. However, depending on the architecture (of both the encoding layers and the trainable layers) these quantities may be limited in the values they can represent. This could significantly limit their ability to represent various functions. This will henceforth be referred to as the *Fourier accessibility* of the quantum architectures.

II. BACKGROUND REVIEW – LINEAR ARCHITECTURES

As discussed in [10], all quantum neural networks that use angle embedding[4] as their encoding strategy produces a truncated Fourier series approximation to the dataset. [10] also specifically explored two families of architectures of quantum neurons: a single-qubit architecture with a series of sequential $\mathcal{SU}(2)$ gates, and a multi-qubit architecture with parallel $\mathcal{SU}(2)$ encoding gates. In this section, the results and the architectures introduced in [10] are explored in depth in Sec. III two alternative neuron architectures are presented with the capability of achieving an exponentially higher Fourier expressivity for the same number of gates.

Consider a quantum neuron that maps a real feature $x \in \mathcal{X}$ onto the quantum circuit via a parametric gate $S(x) = e^{-i\mathcal{G}x}$. In most common architectures the only parametric gates are single qubit rotations $\{R_x, R_y, R_z\}$. For this work, the Pauli-Z generated rotations are used without any loss of generality $\mathcal{G} = \frac{1}{2}\sigma_z = \frac{1}{2}\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$[5], then the embedding gate takes a simple form $S(x) = \left(\begin{smallmatrix} e^{-ix/2} & 0 \\ 0 & e^{ix/2} \end{smallmatrix}\right)$. In general, the dependence of the expected value of any observable on the parameter x is then given by

$$\langle M \rangle = \langle \Psi | \left(S^{\dagger}(x) \otimes \mathbb{1}\right) M \left(S(x) \otimes \mathbb{1}\right) |\Psi\rangle = (c_0 + c_0^*) + c_1 e^{ix} + c_1^* e^{-ix}$$

with some complex parameters c_0 and c_1 which depend on the rest of the circuit and the measurements. This expected value is a function of the feature x with a very simple Fourier series. The *data re-uploading method* [11] is a natural way to construct neurons that give rise to richer Fourier series. These are architectures where several parametric gates depend on the same x. It is the most straightforward to consider gates that have a hardwired dependence on the feature[6]. In particular, such that the expected value of any observable takes the form of a discrete Fourier series

$$f(x, \theta) = \sum_k c_k(\theta) e^{ikx}, \tag{1}$$

where θ the variational parameters and $c_k \in \mathbb{C}$ with $c_{-k}^* = c_k$ for real observables. In Sec. II A and II B, two architectures exhibited in [10] are reviewed. The *Fourier expressivities* of these architectures are of particular interest, that is the list of wavenumbers $\{k_1, k_2, \dots\}$ appearing in the exponents in Eq. (1).

A. Sequential linear

The single-qubit sequential linear method uses repetitions of the same single-qubit encoding gate $S(x)$ interlaced in-between trainable variational layers. Fig. 1a shows this implementation with generalized variational gates. Since the eigenvalues of each unitary are $e^{\pm i\frac{1}{2}x}$, it is straightforward to observe (see e.g. App. A 1) that after n encoding layers the expected value of any observable takes the form

$$f(x, \theta) = \sum_{k=-n}^{n} c_k(\theta) e^{ikx}, \tag{2}$$

Thus, the repetitions have an additive effect such that for n repetitions the final list becomes $\{-n, -n+1, \cdots, 0, \cdots, n-1, n\}$. Each of the wavenumbers in the list gives rise to a sinusoidal term with the same frequency. Therefore, for n repetitions of the encoding $S(x) = e^{-i\mathcal{G}x}$, n distinct Fourier bases are generated.

––––––––

[4] Ref. [25] explores other embedding strategies too, such as basis embedding and amplitude embedding. This work primarily focuses on angle embedding. Nevertheless, it is worth noting that in the circuit model all circuit parameters enter as angles at some level of the description.

[5] extra gates to convert between different generators can be absorbed into the variational gates – see [10]

[6] In principle, one may also consider gates $S(x; \theta) = e^{-if_\theta(x)\mathcal{G}}$ with a *variational* dependence on the feature x, where the parameters f_θ are to be learned. However, this could present additional challenges related to the computation of gradients and increased number of parameters

(**a**) Sequential linear

(**b**) Sequential exponential

(**c**) Parallel linear

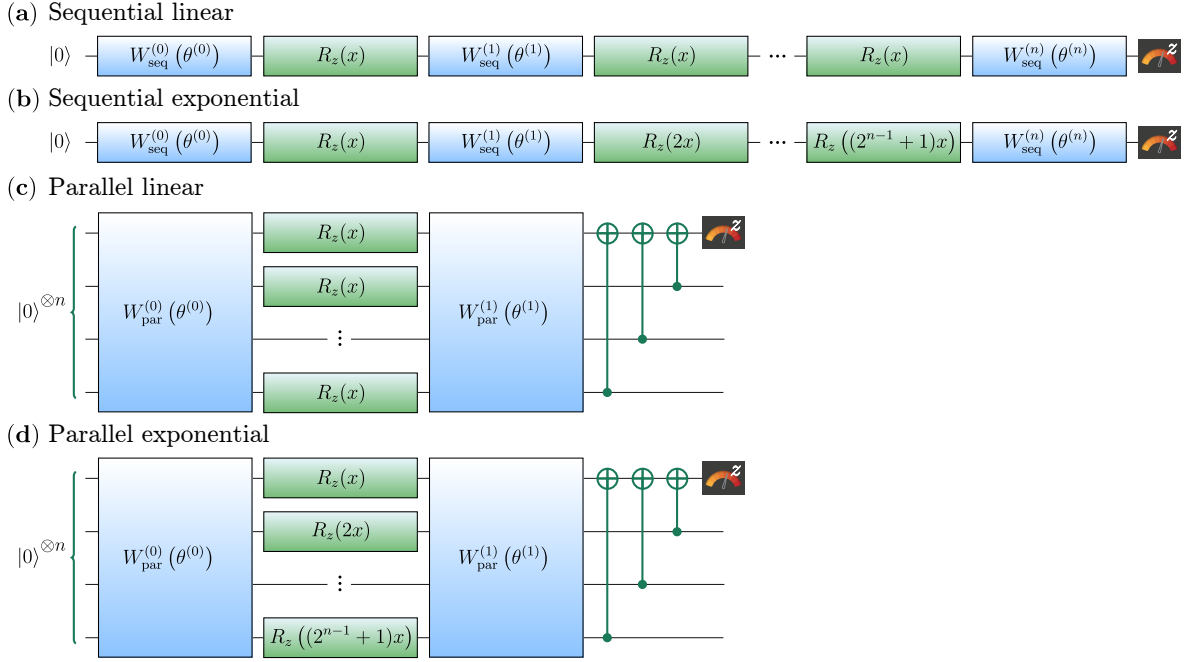
(**d**) Parallel exponential

FIG. 1: The four general circuits under analysis in this paper. In the exponential architectures, the first encoding is kept the same and the subsequent encoding gates are multiplied by the coefficients in Eq. (6). The parallel circuits have a CNOT layer at the end to ensure that all qubits are cooperating in the training by propagating the π-measurement through all quantum wires.

B. Parallel linear

In the parallel setting, the single-qubit encoding gates are applied in parallel on separate qubits – see Fig. 1c. Similarly to the sequential encoding, for n parallel rotations n Fourier bases are produced. This is due to the commutativity between the parallel rotations as they act on separate qubits. The generator $\mathcal{G}$ becomes:

$$\mathcal{G} = \frac{1}{2} \sum_{q=1}^{r} \sigma_z^{(q)}, \tag{3}$$

where q is the qubit index, r indicates the total number of qubits, and $\sigma_z^{(q)}$ is the Pauli-Z matrix applied to the qth qubit. In App. A 2, it is shown that $\mathcal{G}$ – being a square matrix of dimensions 2^r – has $2r + 1$ unique eigenvalues. This suggests a high degree of degeneracy in its eigenspectrum. As before, the subtraction of these values from themselves yields a list of wavenumbers ranging from $-r$ to r generating r Fourier bases.

C. Redundancy

Both in the sequential and parallel linear architectures there is a lot of redundancy in how the feature is encoded into the circuit. This is the easiest to see for the parallel architecture, where most of the eigenvalues of $\exp(ix\mathcal{G})$ are largely degenerate as the encoding commutes with qubit permutations.

III.   RESULTS – EXPONENTIAL ARCHITECTURES

In this section, two new families of architectures are suggested that can encode an exponential number of Fourier bases for a given number of repetitions/parallel encodings. The basis of this generalization is to modify each "subsequent" appearance of the encoding gate in the circuit by a re-scaling of the
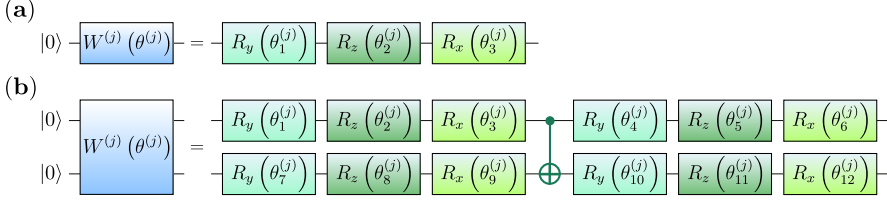
(a)

(b)

FIG. 2: Variational gates used in single – (a) and 2-qubit (b) experiments.

generator $S(x) \to S(mx)$ with an integer m. Keeping the factors m integer guarantees that this procedure results in a discrete Fourier series in the form of Eq. (1).

A. Sequential exponential

It was shown in Sec. II A that the wavenumbers created in the linear models are highly degenerate. By modifying the circuit encoding this degeneracy can be reduced, resulting in adding new wavenumbers to the list. This is accomplished by altering the generators in the individual encoding layers. In the linear case, the diagonal elements of the generator λ_i always belonged to the list $\{-\frac{1}{2}, \frac{1}{2}\}$, but could be altered by scaling the generator $\mathcal{G}$ in each layer. In practice, this is achieved by scaling the embedded data x, and then mathematically associating with the generator. The resultant function becomes

$$f(x, \theta) = \left(W_{1,j_1}^{(0)\dagger} W_{j_1,j_2}^{(1)\dagger} \cdots M_{k',k} \cdots W_{i_2,i_1}^{(1)} W_{i_1,1}^{(0)} \right)$$
$$e^{\left((\lambda_{j_1}^{(1)} + \lambda_{j_2}^{(2)} + \cdots) - (\lambda_{i_1}^{(1)} + \lambda_{i_2}^{(2)} + \cdots) \right) x},$$

where $\lambda_i^{(l)} = a_l \lambda_i = \frac{1}{2}\{-a_l, a_l\}$ for $a_l \in \mathbb{N}$ [7]. In this work, a_l scales as follows $a_l = \{2^0, 2^1, 2^2, \cdots, 2^{n-1} + 1\}$. The motivation behind this choice is the sum of powers of 2, $\sum_{i=0}^{n-1} 2^i = 2^n - 1$, where the largest wavenumber possible, 2^n, is obtained by taking all the positive contributions from the list of eigenvalues, i.e. $k_{max} = \sum_{i=0}^{n-1} 2^i + 2^{n-1} + 1 = 2^n$. Next, one can switch the signs of the positive values to negative starting from the smallest term to produce all integers from -2^n to 2^n. This generates 2^n Fourier frequencies. Fig. 1b shows a quantum circuit encoded using the sequential exponential strategy with 2 layers. App. B demonstrates that this network produces extreme constraints on the Fourier accessibility, and thus is an undesirable choice for general data modelling. However, this scheme motivates extending this idea to parallel architectures.

B. Parallel exponential

To perform this extension, it is appropriate to proceed with a 2-qubit example. The parallel linear method described in Sec. II B produces the generator:

$$\mathcal{G}^{\text{lin}} = \frac{1}{2} \left(\sigma_z \otimes \mathbb{1} + \mathbb{1} \otimes \sigma_z \right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{4}$$

This matrix has 3 unique eigenvalues, $\lambda \in \{-1, 0, 1\}$, and when subtracted from itself – yielding wavenumbers $L_k^{(\text{lin})} = \{-2, -1, 0, 1, 2\}$ – it can produce 2 Fourier bases with frequencies $\{1, 2\}$. One

---

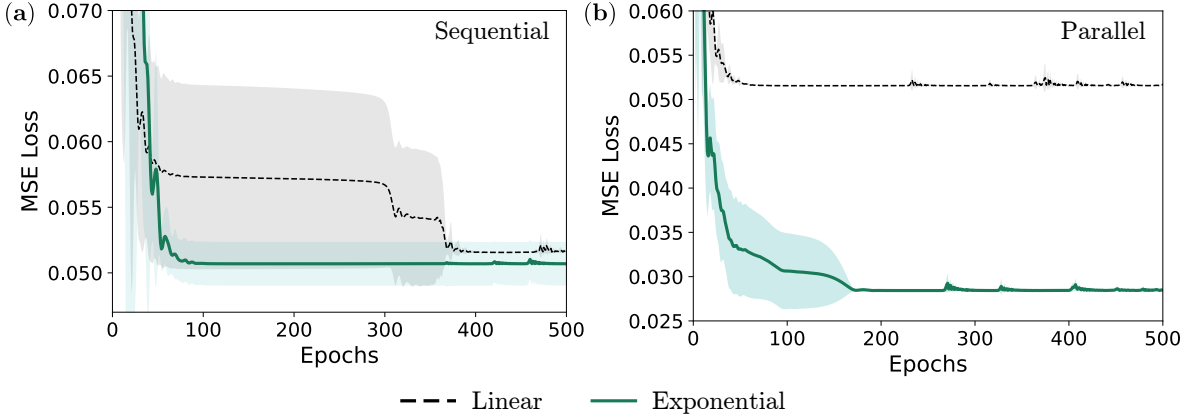[7] In comparison, for linear architectures $a_l = 1$ for all l.

FIG. 3: Training losses indicate a training advantage for the parallel exponential, and that the sequential exponential architecture performs only marginally better than the linear architectures. The training was done on QMware hardware[27] using the PennyLane Python package [28]. The Adam optimizer was employed to minimize a mean squared loss function with a learning rate of $\epsilon = 0.1$ and with uniformly-distributed parameters $\theta \in [0, 2\pi]$.

could generate a matrix with more unique values. For example,

$$\mathcal{G}^{\text{exp}} = \frac{1}{2}\left(3\sigma_z \otimes \mathbb{1} + \mathbb{1} \otimes \sigma_z\right) = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 \end{pmatrix}, \tag{5}$$

is a generator with 4 unique eigenvalues that generate 9 wavenumbers $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$. This generator can be constructed using the quantum circuit shown in Fig. 1d. In this case, a $\mathcal{SU}(4)$ generator is employed. This is decomposed into 2 $\mathcal{SU}(2)$ generators, one using the group parameter, x, and the other $3x$. This can be generalized to n qubits as one can extend the matrix for larger numbers of qubits, i.e. for n qubits $\mathcal{G}$ would be a diagonal matrix starting from $-2^{(n-1)}$ up to $2^{(n-1)}$, producing 2^n Fourier bases. The quantum circuit associated with this generator is an application of Pauli-Z rotations of x with frequencies increasing in the following way:

$$\mathcal{L} = \{2^0, 2^1, 2^2, ..., 2^{n-1} + 1\}, \tag{6}$$

where n is the number of qubits. Note the similarities between the sequential and parallel encodings and their symmetries in the way the circuits are constructed. One also recognizes similarities between the parallel encoding and Kitaev's quantum phase estimation algorithm [26], albeit in this case x is a classical feature.

This can be significantly more expressive than the parallel linear method, but this advantage needs to be accompanied by Fourier accessibility since if the Fourier values of these newly-acquired bases cannot be altered, there would be no advantage in pursuing this setting. Sec. III C shows there is a significant advantage in using parallel exponential encoding in a simple toy example.

C. Training

In this section, the training performance of these 4 architectures on a simple dataset is compared. Each architecture is trained to reproduce a one-dimensional top-hat function. Fig. 4 shows the ground truth as well as the fitting performances of these architectures, and Fig. 3 shows their training performance. It is clear that the parallel exponential architecture fits a closer function to the ground truth, and in contrast, the sequential exponential architecture has the worst performance of all models. Furthermore, the Fourier decompositions of the models in Fig. 6 show that exactly 2 Fourier terms are accessed by the linear architectures and 4 by the exponential ones. Additionally, Fig. 5 demonstrates the performance of the parallel exponential architecture on a trapped ion quantum processor. The IonQ implements a high-fidelity gate-based quantum processing unit through a process known as laser pumping trapped-ions explained in [29]. The hardware was shown to be one of the most accurate in recent benchmarking
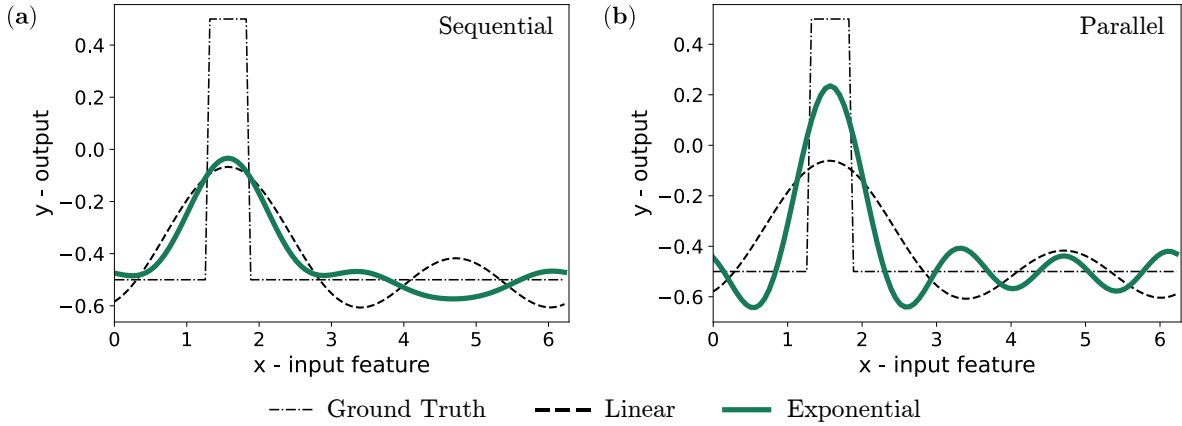
FIG. 4: The QNNs fit the best possible truncated Fourier series on the top-hat function. The parallel exponential architecture provides the best fit, and even though the sequential exponential architecture has access to the same 4 Fourier frequencies, it fails to access all of them efficiently and as a result, it performs sub-optimally. The linear architectures perform similarly to each other, potentially arising from their high Fourier accessibility to the 2 Fourier frequencies that they can represent.
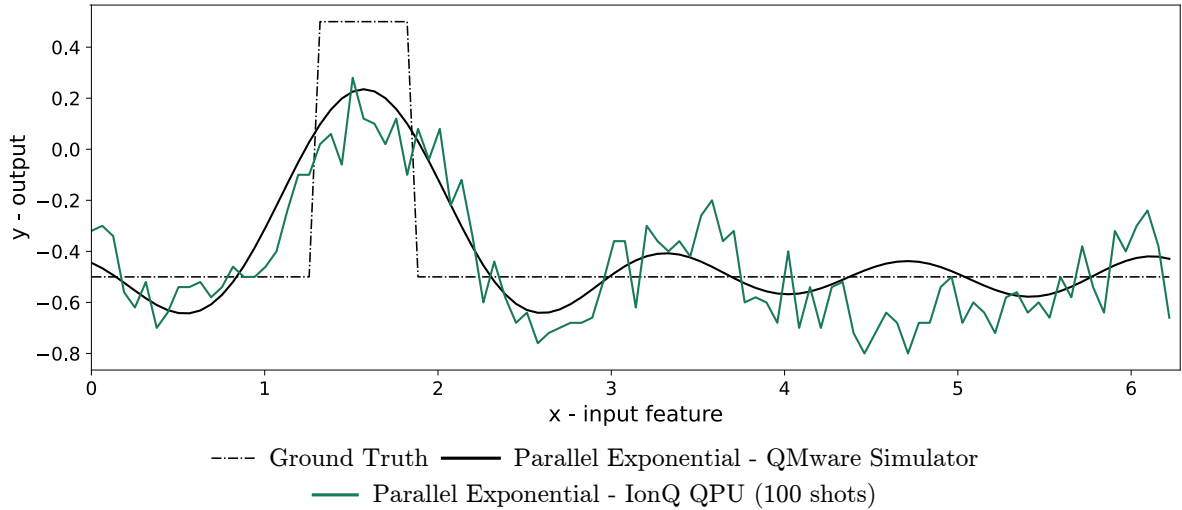

FIG. 5: The parallel exponential fit to the top-hat function on a simulator vs on a quantum processor. The noisy solid line is the evaluation of this network for 100 equally-spaced points using 100 shots.

tests [30]. We specifically used the hardware introduced in [31] with a single-qubit fidelity of 0.997 and a 2-qubit fidelity of 0.9725. The code implementation was done through Amazon Web Services (AWS) Braket, and the process of the forward pass for 100 data points took 4 hours and 11 minutes, due to the delays and queuing times. It can be observed that the low number of shots is the dominant source of noise here and higher shot counts could yield a smoother curve that is closer to the simulator.

D.   Critical Evaluation

While the results for the parallel exponentials are encouraging, it is equally important to understand the limitations of this approach. Firstly, while the exponential growth in the number of Fourier frequencies is evident, this is not the higher limit of Fourier frequency growth. [10] showed that for L repetitions of an encoding gate with a Hilbert space of dimension d, there is an upper limit to this growth of the form
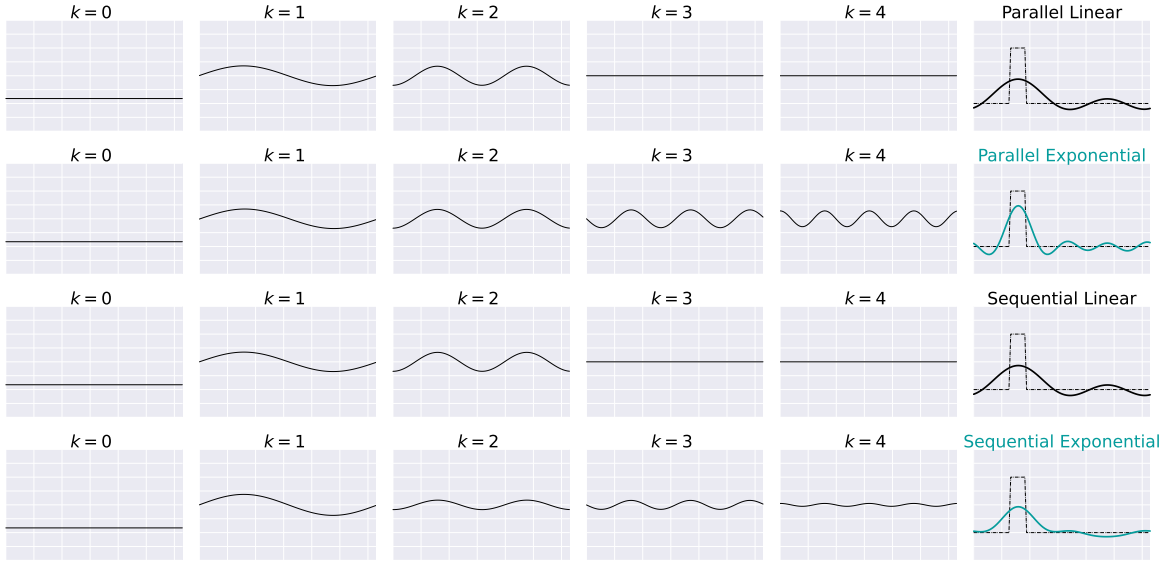
FIG. 6: Fourier decomposition of the four architectures after training to fit the top-hat function. We see that the linear architectures can only access two Fourier frequencies, whereas the exponential ones can access four.

$$K < \frac{d^{2L}}{2} - 1, \tag{7}$$

where K is the number of Fourier frequencies. This suggests that there is a potential for square-exponential growth, whereas the method discussed in this work only grows exponentially. In App. C a mathematical problem is proposed, whose solution could unlock the maximum possible Fourier accessibility.

Secondly, it is important to emphasize that the two parallel architectures are the same with a minor multiplicative factor added in the exponential case. This means that training them for a fixed number of epochs requires the same computational resources. However, adding more Fourier bases by eliminating the degeneracy of the network could result in under-parameterized models. Therefore, it is often necessary to parameterize the exponential architectures more heavily than the linear ones, indirectly affecting the required resources. Every Fourier frequency requires 2 degrees of freedom (real-valued parameters) and an exponentially-growing Fourier space requires the resources to grow exponentially, too. These resources could include the classical memory required to store the parameters or the classical optimizer that needs to calculate the gradient for these parameters. And lastly, extending this to many qubits will still result in barren plateaus.

IV.   CONCLUSION AND FUTURE WORK

This work suggested two new families of QNN architectures, dubbed sequential and parallel exponential circuits, that provided an exponentially growing Fourier space. It was demonstrated that the former struggled with accessing these frequencies, but also that the latter showed an advantage in approximating a top-hat function.

Future work could focus on a quantitative understanding of the Fourier accessibility of these networks, such that the optimal variational parameterization could be chosen for a specific problem. Another possible direction for future work is to depart from hardwired encoding gates. A natural elementary step in this direction is to consider single-qubit gates of the form $S_i(x, w_i) = \exp(-i\, x w_i \frac{1}{2}\sigma_z)$, where the scaling factor w_i is an independent scalar trainable parameter for each occurrence of the encoding gate in the circuit. In this case, the final wavevectors k are linear combinations of the parameters w_i

that can be potentially trained efficiently.

[1] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1:403–409, 2021.

[2] Michael Perelshtein, Asel Sagingalieva, Karan Pinto, Vishal Shete, et al. Practical application-specific advantage through hybrid quantum computing. *arXiv preprint arXiv:2205.04858*, 2022.

[3] N Schetakis, D Aghamalyan, P Griffin, and M Boguslavsky. Review of some existing qml frameworks and novel hybrid classical–quantum neural networks realising binary classification for the noisy datasets. *Scientific Reports*, 12(1):1–12, 2022.

[4] Asel Sagingalieva, Andrii Kurkin, Artem Melnikov, Daniil Kuhmistrov, et al. Hyperparameter optimization of hybrid quantum neural networks for car classification. *arXiv preprint arXiv:2205.04878*, 2022.

[5] Asel Sagingalieva, Mohammad Kordzanganeh, Nurbolat Kenbayev, Daria Kosichkina, Tatiana Tomashuk, and Alexey Melnikov. Hybrid quantum neural network for drug response prediction. *arXiv preprint arXiv:2211.05777*, 2022.

[6] Maria Schuld and Nathan Killoran. Is quantum advantage the right goal for quantum machine learning? *arXiv preprint arXiv:2203.01340*, 2022.

[7] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103:150502, 2009.

[8] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5:1–7, 2014.

[9] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*, 2010.

[10] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physics Reviews A*, 103(3):032430, 2021.

[11] Adriá n Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

[12] Mohammad Kordzanganeh, Aydin Utting, and Anna Scaife. Quantum machine learning for radio astronomy. *arXiv reprint arXiv:2112.02655*, 2021.

[13] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12:1–11, 2021.

[14] Christoph Spengler, Marcus Huber, and Beatrix C. Hiesmayr. Composite parameterization and haar measure for all unitary and special unitary groups. *Journal of Mathematical Physics*, 53(1):013501, 2012.

[15] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3), 2019.

[16] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80:012304, 2009.

[17] Aram W. Harrow and Richard A. Low. Random quantum circuits are approximate 2-designs. *Communications in Mathematical Physics*, 291:257–302, 2009.

[18] Andris Ambainis and Joseph Emerson. Quantum t-designs: t-wise independence in the quantum world. In *Twenty-Second Annual IEEE Conference on Computational Complexity*, pages 129–140, 2007.

[19] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9:1–6, 2018.

[20] Andrea Skolik, Jarrod R. McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. Layerwise learning for quantum neural networks. *Quantum Machine Intelligence*, 3:1–11, 2021.

[21] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3:214, 2019.

[22] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12:1–12, 2021.

[23] Chen Zhao and Xiao-Shan Gao. Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus. *Quantum*, 5:466, 2021.

[24] Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.

[25] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.

[26] Alexei Y. Kitaev. Quantum measurements and the abelian stabilizer problem. *Electronic Colloquium on Computational Complexity*, TR96, 1996.

[27] QMware. QMware — The first global quantum cloud. https://qm-ware.com/, 2022.

[28] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[29] Philipp Schindler, Daniel Nigg, Thomas Monz, Julio T. Barreiro, Esteban Martinez, Shannon X. Wang, Stephan Quint, Matthias F. Brandl, Volckmar Nebendahl, Christian F. Roos, Michael Chwalla, Markus Hennrich, and Rainer Blatt. A quantum information processor with trapped ions. *New Journal of Physics*, 15:123012, 2013.

[30] Mohammad Kordzanganeh, Markus Buchberger, Maxim Povolotskii, Wilhelm Fischer, Andrii Kurkin, Wilfrid Somogyi, Asel Sagingalieva, Markus Pflitsch, and Alexey Melnikov. Benchmarking simulated and physical quantum processing units using quantum and hybrid algorithms. *arXiv preprint arXiv:2211.15631*, 2022.

[31] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10:1–6, 2019.

Appendix A: Fourier estimation of the linear architectures

1. Sequential linear

To see the extraction of the Fourier modes more explicitly, one can write the function represented by this circuit as the expectation value of a measurement operator M:

$$f(x,\theta) = \langle 0| W^{(0)\dagger}(\theta)S^\dagger(x)W^{(1)\dagger}(\theta)\cdots M\cdots W^{(1)}(\theta)S(x)W^{(0)}(\theta)|0\rangle \tag{A1}$$

$$= \langle 0| W^{(0)\dagger}(\theta)e^{i\mathcal{G}x}W^{(1)\dagger}(\theta)\cdots M\cdots W^{(1)}(\theta)e^{-i\mathcal{G}x}W^{(0)}(\theta)|0\rangle \tag{A2}$$

Writing this in tensor notation yields

$$= W_{1,j_1}^{(0)\dagger}e^{i\mathcal{G}_{j_1,j_2}x}W_{j_2,j_3}^{(1)\dagger}\cdots M_{k',k}\cdots W_{i_3,i_2}^{(1)}e^{-i\mathcal{G}_{i_2,i_1}x}W_{i_1,1}^{(0)}, \tag{A3}$$

where $W = W(\theta)$ and the index summation convention is employed. We can simplify this expression further by employing

$$e^{-i\mathcal{G}_{i_2,i_1}x} = e^{\lambda_{i_1}x}\delta_{i_1,i_2},$$

where $\lambda_{i_1} = \{-\frac{1}{2}, \frac{1}{2}\}$ represents the eigenvalues of $\mathcal{G}$. This simplifies our function:

$$= \left(W_{1,j_1}^{(0)\dagger}W_{j_1,j_2}^{(1)\dagger}\cdots M_{k',k}\cdots W_{i_2,i_1}^{(1)}W_{i_1,1}^{(0)}\right)e^{\left((\lambda_{j_1}+\lambda_{j_2}+\cdots)-(\lambda_{i_1}+\lambda_{i_2}+\cdots)\right)x}. \tag{A4}$$

This means that for every combination of $\{\{i_1,\cdots\}, \{j_1,\cdots\}\}$, the QNN produces a wave-front with a wavenumber[8] $k = (\lambda_{j_1} + \lambda_{j_2} + \cdots) - (\lambda_{i_1} + \lambda_{i_2} + \cdots)$. Note that for n repetitions of this encoding, one obtains 2^{2n} combinations for k, but this also includes a high level of degeneracy for each resultant wavenumber. Specifically, [12] showed that the degeneracy of the wavenumber of this circuit is given by $\{\binom{2n}{2n}, \binom{2n}{2n-1}, \binom{2n}{2n-2}, \cdots, \binom{2n}{0}\}$ for the wavenumbers $\{-n, -(n-1), \cdots, -1, 0, 1, \cdots, n-1, n\}$. Keeping this in mind, one can rewrite Eq A4 as

$$f(x,\theta) = \sum_{k=-n}^{n} c_k(\theta)e^{ikx} \tag{A5}$$

$$= c_0 + \sum_{m=1}^{n} 2|c_m|\cos(mx - arg(c_m)) \tag{A6}$$

$$= c_0 + 2\sum_{m=1}^{n} c_m^r\cos(mx) - c_m^i\sin(mx), \tag{A7}$$

where c_m are complex coefficients and $(c_m^r, c_m^i) = (Re(c_m), Im(c_m))$.

2. Parallel linear

A parallel linear encoding consists of a 2-qubit system where a Pauli-Z rotation of x is applied to each qubit. This means that two encodings are to be considered: the first encoding considers only the rotation applied to the first qubit making its encoding generator $\mathcal{G}^1 = \sigma_z \otimes \mathbb{I}$ and the second $\mathcal{G}^2 = \mathbb{I} \otimes \sigma_z$:

$$\mathcal{G}^1 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \ \mathcal{G}^2 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{A8}$$

───────

[8] Note that this happens at the level of the density matrices and is independent of the choice of M.
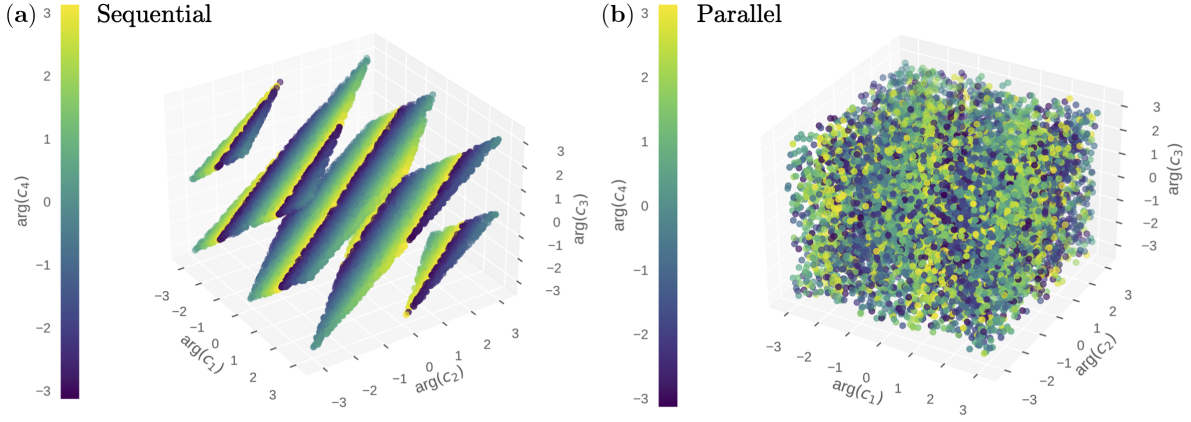
FIG. 7: Fourier phases of the two exponential architectures: (a) sequential and (b) parallel. Each architecture was realised 10,000 times and their arguments were calculated by taking the discrete Fourier transform of their outputs. We see that the sequential architecture has a restricted 4-dimensional behavior and that there seems to exist a linear dependence between the phases, demonstrating a lack of Fourier accessibility. In contrast, the parallel architecture can fill the space, but still, some constraint is visible between the $\arg(c_1)$ and $\arg(c_4)$ bases.

Since the gates are applied to separate qubits, the generators commute with each other, and since the group parameters applied to each gate are the same, the elements of the encoding matrix can be written as:

$$\left[e^{i\mathcal{G}^1 x}e^{i\mathcal{G}^2 x}\right]_{mn} = e^{i\lambda_j^1 x}e^{i\lambda_k^2 x}\delta_{mj}\delta_{kn} = e^{i(\lambda_j^1+\lambda_k^2)x}\delta_{mj}\delta kn = e^{i\mathcal{G}_{mn}^{\mathrm{comb}}x}, \tag{A9}$$

where $\mathcal{G}^{\mathrm{comb}}$ is the combined generator, and δ is the Kronecker delta which acts as the identity matrix in index notation. Using this expression, one could write

$$\mathcal{G}_{mn}^{\mathrm{comb}} = \mathcal{G}^1 + \mathcal{G}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{A10}$$

Similarly to the calculations in App. A 1, this generator provides us with a list of eigenvalues $\lambda \in \{-1, 0, 1\}$, which when subtracted from itself yields 2 Fourier frequencies and 5 wavenumbers $k \in \{-2, -1, 0, 1, 2\}$. This calculation can be generalized to any number of qubits, and the number of Fourier frequencies increases linearly with the number of qubits.

Appendix B: Constraints of the sequential exponential

In Sec. III C it was shown that both sequential and parallel exponential architectures represented four Fourier frequencies, but that the latter achieved a lower training loss and a better fit to the top-hat function. This is due to the reduced Fourier accessibility of the sequential architecture, meaning that it lacks the freedom to achieve any desired point in the Fourier space. The models with 4 Fourier frequencies are realized in a 9-dimensional space that includes c_0, and the real and imaginary values of c_i. Each realization of the trainable parameters of the quantum circuit produces a 9-dimensional array creating a point in this 9-dimensional space. Realizing these two architectures many times makes it possible to analyze the geometry of the Fourier space for each architecture, but for manual observation, it is important to find an efficient way to reduce this dimensionality to 3 dimensions (or 4 with color). Fig. 7 shows a choice for this dimensionality reduction by investigating the arguments of the complex Fourier coefficients which based on Eq A7 represent the phases of the co-sinusoidal terms. These show that the sequential exponential architecture is dramatically constrained in the collection of phases it can represent and that the parallel exponential is unconstrained in this way.

Appendix C: Beyond exponential growth

As in App. A 1, to reach the final list of wavenumbers needed to subtract the eigenvalues of the Hamiltonian in pairs. In this section, a mathematical problem is proposed that can lead to the highest possible Fourier series.

Problem 1 *For a given $m \in \mathbb{N}$, find a list of integers $L \in \mathbb{Z}^m$ such that when subtracted from itself, it produces a new list $k_L^{(max)} = \{x - y \mid x, y \in L\}$ whose elements are sequential integers and, except for zero, all the elements have a degeneracy of exactly 1.*

The problem statement produces a list of eigenvalues L from which one can produce a list of wavenumbers $k_L^{(max)}$. After finding this list, it is crucial to check if one can create a diagonal Hamiltonian using R_Z rotations and non-parameterized gates whose diagonal elements are the numbers in L.