QML-Essentials—A Framework for working with Quantum Fourier Models

Melvin Strobl[*] ⬤
Karlsruhe Institute of Technology
Karlsruhe, Germany
melvin.strobl@kit.edu

Maja Franz[*] ⬤
Technical University of
Applied Sciences Regensburg
Regensburg, Germany
maja.franz@othr.de

Eileen Kuehn ⬤
Karlsruhe Institute of Technology
Karlsruhe, Germany
eileen.kuehn@kit.edu

Wolfgang Mauerer ⬤
Technical University of
Applied Sciences Regensburg
Siemens AG, Technology
Regensburg/Munich, Germany
wolfgang.mauerer@othr.de

Achim Streit ⬤
Karlsruhe Institute of Technology
Karlsruhe, Germany
achim.streit@kit.edu

Abstract—**In this work, we propose a framework in the form of a Python package, specifically designed for the analysis of Quantum Machine Learning models. This framework is based on the PennyLane simulator and facilitates the evaluation and training of Variational Quantum Circuits. It provides additional functionality ranging from the ability to add different types of noise to the classical simulation, over different parameter initialisation strategies, to the calculation of expressibility and entanglement for a given model. As an intrinsic property of Quantum Fourier Models, it provides two methods for calculating the corresponding Fourier spectrum: one via the Fast Fourier Transform and another analytical method based on the expansion of the expectation value using trigonometric polynomials. It also provides a set of predefined approaches that allow a fast and straightforward implementation of Quantum Machine Learning models. With this framework, we extend the PennyLane simulator with a set of tools that allow researchers a more convenient start with Quantum Fourier Models and aim to unify the analysis of Variational Quantum Circuits.**

Index Terms—**Quantum Machine Learning, Quantum Computing, Quantum Software Framework**

I. Introduction

Variational Quantum Circuit (VQC) are a promising approach on the Noisy Intermediate Scale Quantum (NISQ) era towards Fault-Tolerant Quantum Computing (FTQC). With pervasive challenges such as barren plateaus [1] and the resulting limitations on trainability, there are still many questions to be answered [2]. It was found that VQCs following a certain structure can be represented by a truncated Fourier series [3]. In the remainder of this paper we will refer to such models as Quantum Fourier Model (QFM) (which are different from the well-known Quantum Fourier Transform). The Fourier spectrum is considered as an important property to characterise a Quantum Machine Learning (QML) model, as it reflects its ability to learn non-linearities with asymptotically being a

universal function approximator [3]. The study of the spectrum of QFMs sparked a series of exciting research [4]–[7] and gives insights on the dequantisability of a given QML model [8], also in the context of noisy computation [9].

By introducing QML-Essentials we provide a tool to explore the properties of QFMs in the context of QML. We also implement algorithms to compute key metrics of VQCs, such as expressibility and entangling capability [10]. Since noise is an intrinsic property of Quantum Computing (QC), QML-Essentials provides the ability to apply noise to the simulation to investigate the behaviour of such algorithms on today's NISQ devices. By providing open access to the algorithms, we aim to standardise the computation of these metrics for specific algorithms across the QML community [11].

In this article, we first discuss related work and other quantum frameworks for working with QML models in Sec. II. We then give an overview of the framework, the implemented algorithms and metrics in Sec. III. In Sec. IV, we show some examples of how the framework can be used to reproduce existing results in the literature and conclude in Sec. V.

II. Related Work

We acknowledge that there are numerous other exceptional software projects that facilitate research in the field of QC [12].

The PennyLane [13] library, which functions as a backend for the proposed framework, offers a classical simulator of quantum computing and a comprehensive suite of tools that extend the capabilities of the simulator. These include the ability to simulate noise, calculate gradients, and perform numerous other operations. Nevertheless, while PennyLane also provides an implementation to calculate the Fourier coefficients of a given circuit using the Fast Fourier Transform (FFT), it is too limited for extensive research with QFMs.

---

[*]Equal contribution

With PennyLane being more focused on hybrid computation and machine learning, the framework Qiskit [14] is targeted towards more hardware-aware development.

In comparison to PennyLane, Qiskit offers a more extensive suite of tools for direct manipulation of quantum circuits during the transpilation process, as well as for direct manipulation at the pulse level. However, it should be noted that Qiskit provides limited support for QML models compared to PennyLane, which restricts its applicability to QFMs.

Horqrux, a backend of Qadence [15], is a simulation framework designed for QML. It provides options to fit non-linear functions as well as solving partial differential equations. There are other full-stack frameworks such as Qibo [16] and QRISP [17] that provide different simulator backends and allow for working with QML in general to some extent, but are not tailored to QFM.

To the best of our knowledge, no framework unifies the analysis of QFMs in the context of QML with the variety of tools that are elaborated in this paper.

III. QML-ESSENTIALS FRAMEWORK

The modules of the Python package, which is available on PyPi[1] and Github[2], are summarised in Tab. I with an overview of its dependencies shown in Fig. 1. The `Model` sits at the core of QML-ESSENTIALS and is built upon a chosen Ansatz. It can either be used in training and other applications outside of QML-ESSENTIALS, or passed to the corresponding functions of the `Expressibility` (Sec. III-C1), `Entanglement` (Sec. III-C2 and Sec. III-C3), `Coefficients` (Sec. III-B1) and `FourierTree` (Sec. III-B2) modules explained in the referenced sections. In this work, we only focus on the aforementioned aspects of our framework. Nonetheless it should be noted that there is a plethora of other features in QML-ESSENTIALS such as (1) different initialisation strategies and parameter sampling, (2) changing of, or providing custom feature maps, (3) different measurements based on provided output shapes, (4) caching of results using hashed parameters and (5) oversampling of the Fourier spectrum. These features are not covered in this article but are explained on our curated documentation page.[3]

TABLE I
PYTHON MODULE OVERVIEW.

| Module	Description
Model	Data-reuploading model class with various options to change initialisation strategy, encoding, measurement qubit and noise.
Ansaetze	Set of circuits to be used within a model.
Coefficients	Calculate coefficients (FFT or analytical).
Expressibility	Tools to calculate expressibility.
Entanglement	Calculate entangling capability (Meyer-Wallach measure or Bell measurements).
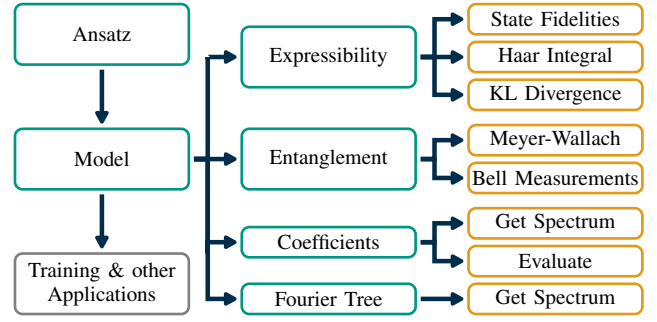
Fig. 1. Overview of the classes of the QML-ESSENTIALS framework. Modules of the framework are depicted in teal, an excerpt of their corresponding functions in orange.

In the following, we introduce the main concepts to QML models, and explain how they can be utilised and analysed using modules of the QML-ESSENTIALS framework.

A. Quantum Machine Learning

Generally, the goal of QML is the same as for classical machine learning, that is to "learn" a function f, using a function approximator f_θ, parametrised by p parameters $\theta \in \mathbb{R}^p$. The parameters θ are "trained" to minimise the difference between $f(x)$ and $f_\theta(x)$, for some input $x \in \mathbb{R}^N$ with N input features, using classical optimisation routines, such as gradient descent [18].

In the case of QML, the function approximator utilises a VQC, characterised by the parametrised unitary U_θ, acting on a system of n qubits. The function evaluation of the approximator then corresponds to the expectation value of an observable $\mathcal{O}$ on the circuit's state:

$$f_\theta(x) = \langle 0|^{\otimes n} U_\theta^\dagger(x) \mathcal{O} U_\theta(x) |0\rangle^{\otimes n}. \tag{1}$$

B. Quantum Fourier Models

The basis of our framework forms the QFM, which utilises a specific structure of unitary U_θ, namely an interleaving pattern of trainable- and encoding unitaries $W := W_\theta$ and S respectively:

$$U_\theta(x) = W^{(L+1)} S(x) W^{(L)} \cdots W^{(2)} S(x) W^{(1)}. \tag{2}$$

As shown in the seminal works of Ref. [19] and Ref [3], circuits following this unitary structure (1) represent universal function approximators [19] and (2) can be represented as a truncated Fourier series with the frequencies $\omega \in \Omega$ and their corresponding magnitudes $c_\omega(\theta)$ as its characteristic properties:

$$f_\theta(x) = \sum_{\omega \in \Omega} c_\omega(\theta) e^{i\omega x}. \tag{3}$$

1) Coefficient calculation using the FFT: Practically, basic signal analysis allows us to retrieve the coefficients of a given circuit by evaluating its expectation value Eq. 1 at different inputs and applying a FFT. Given a model, its coefficients can be estimated using the static `Coefficients.get_spectrum()` method.

While providing a fast and in general reliable method, the estimation of coefficients using the FFT also bears some disadvantages. Firstly, the frequencies must be evenly spaced or the number of sampling points must be chosen such that intermediate frequencies are captured correctly. Even then it only gives an approximation based on the provided sampling points without taking into account the actual circuit properties. This can lead to scenarios in which frequencies are not captured at all [4]. Secondly, the highest frequency must be estimated in advance to fulfil the Nyquist criterium. Otherwise frequency artefacts caused by the repeating structure of the Fourier transform will be observed.

2) Coefficient calculation using the analytical method: Wiedmann *et al.* [4] introduced an analytical method to estimate the coefficients of a given QML model. The proposed algorithm builds upon the work by Nemkov *et al.* [6], which proposes an expansion of $f_\theta(x)$ in terms of trigonometric polynomials. This method relies on all operations in the circuit being either Pauli-rotation or Clifford gates, to which any VQC can be decomposed into. The circuit U_θ is then transformed into a circuit that only consists of Pauli rotation gates, with all Clifford gates moved towards the end of the circuit to be included in the observable. We implement their method in our framework and it is accessible via the `FourierTree.get_spectrum()` method after instantiating the `FourierTree` class using a QML model. While significantly slower, it provides an accurate estimation of the Fourier spectrum which leaves the choice between these two methods up to the user.

C. Expressibility and Entangling Capability

As the design of Ansätze in QML is still an open field of research, our framework provides tools to calculate the expressibility and entangling capability as two common metrics of interest.

1) Expressibility: For the expressibility we utilise the definition introduced in Ref. [10] which is the Kullback-Leibler (KL) divergence [20] between the distributions obtained by sampling from the Haar integral $\int_{\text{Haar}} (|\psi\rangle\langle\psi|)^{\otimes t} d\psi$ and the model $\int_\Theta (|\psi_\theta\rangle\langle\psi_\theta|)^{\otimes t} d\theta$ respectively:

$$D_{\text{KL}}\left(\hat{P}_{\text{Model}}(F;\boldsymbol{\theta}) \| P_{\text{Haar}}(F)\right) \qquad (4)$$

Here, the fidelity $F = |\langle\psi_\varphi | \psi_\phi\rangle|$ is the probability of state overlaps, whereas the distributions of state overlaps is $p(F = |\langle\psi_\varphi | \psi_\phi\rangle|)$.

This metric yields zero if $\hat{P}_{\text{Model}}(F;\boldsymbol{\theta}) = P_{\text{Haar}}(F)$, meaning the states sampled from the QFM are Haar distributed. For the least expressive case, i.e. the idle circuit, the KL divergence becomes $\ln(n_{\text{bins}})$ where n_{bins} describes the number of bins that are used for discretising the probability distribution using a histogram. In this work, we refer to the expressibility as the inverse of KL divergence.

In QML-ESSENTIALS, the expressibility of a given model can be calculated with the following steps: (1) The state fidelities of two uniformly random parameter sets are calculated using the corresponding method providing a number of
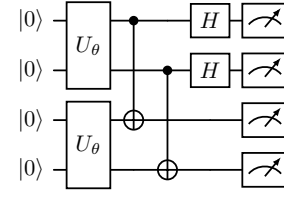

Fig. 2. Setup of a "Bell-Measurement" for a 2 qubit circuit described by U_θ.

samples, (2) the Haar integral for the same number of qubits as in the model is computed, (3) the results of both calculations are passed to the `kullback_leibler_divergence` method of the `Expressibility` class to obtain the distance between the two distributions given a number of bins.

2) Entangling Capability using Meyer-Wallach measure: As shown in Refs. [21], [22], the entangling capability for a n-qubit system can be defined based on the trace of the squared partial density matrix ρ_k for subsystem k:

$$Q(|\psi\rangle) = 2\left(1 - 1/n \sum_{k=0}^{n-1} \text{Tr}\left[\rho_k^2\right]\right) \qquad (5)$$

This metric has the property that if $\text{Tr}\left[\rho_j^2\right] = 1 \quad \forall j$, implying $Q = 0$, then $|\psi\rangle$ is a product state whereas $Q = 1 \iff \text{Tr}\left[\rho_k^2\right] = 1/2 \quad \forall k$, meaning the state is maximally mixed. Notable, access to the density matrix is required for this metric, which is not available on real devices without further ado.

In QML-ESSENTIALS this metric can be calculated using the `meyer_wallach` method within the `Entanglement` module class, given a number of parameter samples.

3) Entangling Capability using Bell-Measurements: As an alternative method to calculate the entangling capability, we implement the "Bell-Measurement" [23]–[25]. We consider a circuit where the state of interest is vertically prepared twice and an inverse Bell-state is applied onto each pair of qubits between the two subsystems. The setup is depicted in Fig. 2.

It was shown that the squared trace of the density matrix is linearly dependent on the probability of measuring the parity between each of the qubits in the individual subsystems:

$$\text{Tr}\left[\rho_k^2\right] = 1 - 2 \cdot P_{\text{odd},k}, \qquad (6)$$

where $P_{\text{odd},k}$ is the probability of odd, non-zero parity in the outcomes of the kth qubit on each copy. Inserting this in Eq. 5, translates to

$$Q(|\psi\rangle) = 2\left(1 - \frac{1}{n} \sum_{k=0}^{n-1} (1 - 2 \cdot P_{\text{odd},k})\right), \qquad (7)$$

resulting in the same estimate of the entangling capability as when using the squared trace of the density matrix directly.

The `bell_measurements` method within the `Entanglement` in QML-ESSENTIALS provides a way to calculate the Bell-measurement, given a number of parameter samples.

While the computational costs of the "Bell-Measurement" also scale exponentially with the number of qubits in a classical simulation of the QML model, it provides a physical observable measurement of the entangling capability, applicable to real quantum systems.

D. Noise

As long as FTQCs are not available, noise remains one of the primary challenges of QML, not only distorting model predictions, but also limiting the trainability [1]. As the effect of noise on QFM is an interesting research direction [9], we offer a low-barrier interface to enable various types of noise as shown in Tab. II to be added to a QML model.

TABLE II
AVAILABLE NOISE TYPES IN QML-ESSENTIALS.

Noise	Description
BitFlip	Per-gate bit-flip with probability p_{bf}.
PhaseFlip	Per-gate phase-flip with probability p_{pf}.
Depolarization	Per-gate depolarisation with probability p_{dp}.
AmplitudeDamping	Amplitude Damping noise with probability p_{ad} applied at the end of the circuit.
PhaseDamping	Phase Damping noise with probability p_{pd} applied at the end of the circuit.
ThermalRelaxation	Thermal relaxation of the system characterised by t1, t2 and gate time factor t_factor.
Measurement	Bit-Flip error with probability p_{me} applied at the very end of the circuit.
StatePreparation	Bit-Flip error with probability p_{sp} applied at the very beginning of the circuit.
GateError	Imprecise gate operations with error $\epsilon \sim \mathcal{N}(0, \mu)$ as coherent per-gate noise.

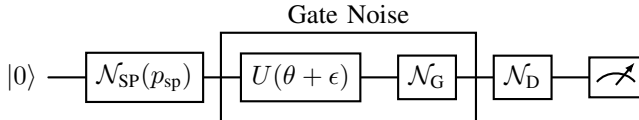Our parameterisable noise model is depicted in Fig. 3 for a single qubit model.

Fig. 3. Noise model consisting of state preparation noise $\mathcal{N}_{SP}$, coherent rotational error ϵ that goes into a noisy gate U with incoherent gate error $\mathcal{N}_G$, damping noise $\mathcal{N}_D$.

Fig. 4 shows the decomposition of the gate noise $\mathcal{N}_G$ into bit-flip, phase-flip and depolarisation. Similarly, Fig. 5 shows the decomposition of the incoherent gate error $\mathcal{N}_D$ into amplitude damping and phase damping noise, as well as the measurement error, each of which is parametrised by its corresponding probability.
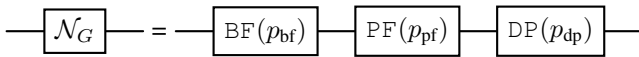

Fig. 4. Decomposition of a single noise operation applied after each gate into bit-flip, phase-flip and depolarising noise.

The individual types of incoherent noise, summarised in Tab. II utilise the Kraus-operator mechanism implemented
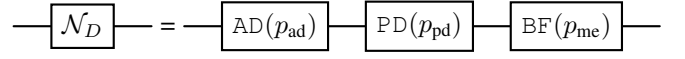

Fig. 5. Decomposition of a noise operation applied at the end of the circuit into amplitude- and phase damping noise.
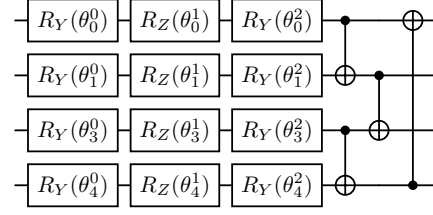

Fig. 6. A single layer of a circular 4-qubit Hardware Efficient Ansatz.

in PennyLane [13]. As StatePreparation and Measurement errors are not directly implemented in PennyLane, we model these by applying a bit-flip Kraus channel at the beginning, or end of the circuit, respectively. Additionally, the coherent GateError, in which $\epsilon \sim \mathcal{N}(0, \mu)$ is drawn randomly from a Gaussian distribution for each gate, allows for modelling unintended rotation offsets on potential real devices.

For specific noise parameter tuning to represent actual quantum systems, we refer to Ref. [26] for a review. By providing a noise_params Python dictionary to the call of the Model class, the full parametrised noise model can be applied to the corresponding VQC.

E. Ansätze

QML-ESSENTIALS includes a set of predefined Ansätze that can be used to build up a model. Currently, this set includes • Circuit 1*, • Circuit 2*, • Circuit 3*, • Circuit 4*, • Circuit 6*, • Circuit 9*, • Circuit 10*, • Circuit 15*, • Circuit 16*, • Circuit 17*, • Circuit 18*, • Circuit 19*, • No Entangling, • Strongly Entangling, • Hardware Efficient,. Ansätze marked with * are implemented based on the Work from Sim *et al.* [10] and can be viewed in their corresponding paper.

For the Hardware-Efficient Ansatz, which only utilises native gates of a typical quantum device and linearly scaling circuit depths, we chose the structure as shown in Fig. 6. We acknowledge that this approach is sometimes realised without the last CNOT gate, which we included for symmetric reasons.

The Strongly-Entangling Ansatz is inspired by [27] and displayed in Fig. 7.

As data embedding we utilise a Pauli R_X rotation as default. Further embeddings and Ansätze can be added by providing a callable function upon instantiation of the model.

IV. EXAMPLES AND VALIDATION

In this section, numerical results are presented that validate the implementation of the coefficients, expressibility and entanglement calculation using a subset of the Ansätze presented in Ref. [10].
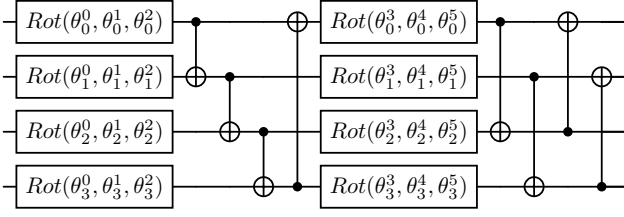
Fig. 7. A single layer of a 4-qubit Strongly Entangling Ansatz as introduced in Ref. [27].

Notable, the results presented here show only a fraction of what is covered by automated testing in our continuous integration and development pipeline. For all of the following numerical results we use a model with 4 qubits and a single variational layer if not stated otherwise. The parameters of the model are sampled from a uniform distribution between 0 and 2π, with 200 samples in the case of the coefficients and 5000 samples for the expressibility and entanglement calculations.

A. Coefficients

Although no reference results are available for the circuits introduced in Ref. [10], we can validate our implementation of the coefficients by comparing the analytical calculation with the results obtained using the FFT. In this experiment we focus on the measurement of a single qubit. The upper part of Fig. 8 shows the results of this validation, where each measurement represents the average of the coefficients over all frequencies. As mentioned in Sec. III-B1, the calculation of coefficients using the FFT bears some caveats when it comes to the estimation of the correct number of frequencies. We demonstrate this discrepancy in the lower part of Fig. 8 where it is clearly visible that only a few circuits actually contain the frequencies indicated by the FFT. By filtering out the frequencies that are not present in the circuit, we obtain identical results from both methods.
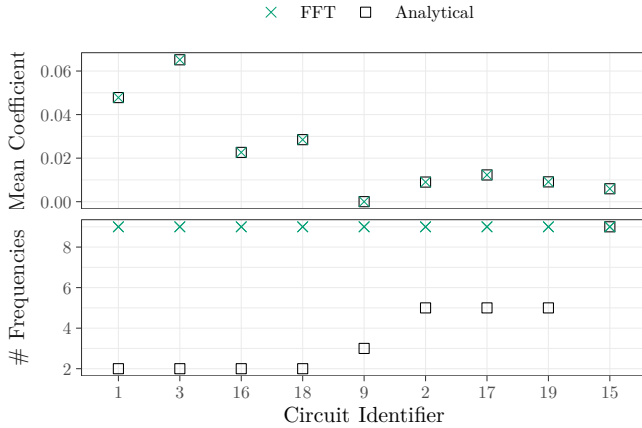

Fig. 8. Top: Fourier coefficients mean values of a circuit obtained using the analytical method and the FFT. Bottom: Difference between the two methods in terms of the number of frequencies.

B. Expressibility

The validation of our implementation of the expressibility is performed by referring to the results from the study by Sim *et al.* [10]. Fig. 9 presents the outcomes of this validation, with expressibility defined as the inverse of the KL divergence between the Haar distribution and the model distribution. It is evident that there is a near-perfect alignment between the reference results and the experimental findings.
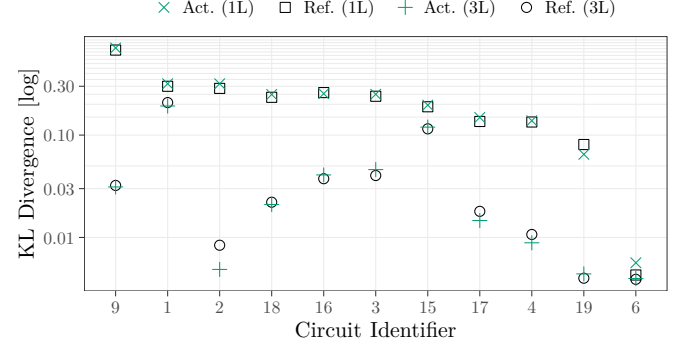

Fig. 9. Expressibility of various circuit with respect to the Haar distribution as KL divergence as reference from [10] (black) and our implementation (teal) for one and three layers.

C. Entangling Capability

Similarly to expressibility, we validate our implementation of calculating the entangling capability using the results from Ref. [10] as a reference with the exact same setup. Fig. 10 shows the results of this validation. It can be observed that the entangling capability is generally overestimated with an increasing discrepancy towards lower values as well as a higher number of layers. The discrepancy may be partly attributed to the unavailability of the precise results from the experiments in Ref. [10]. Furthermore, the present implementation is based on the work of Brennen *et al.* [22]. While this is technically equivalent, it is not exactly the same as the method used in Sim *et al.* [10]. However, it is notable that the results of both methods, the Meyer-Wallach and the Bell-Measurements, align perfectly.
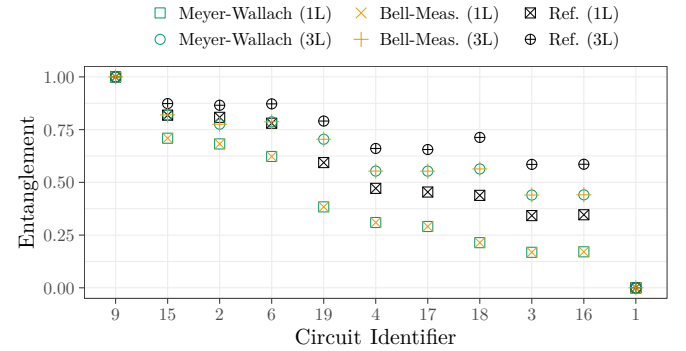

Fig. 10. Entangling Capability of various circuits as reference from [10] (black) and our implementation using the Meyer-Wallach (teal) measure and Bell-Measurements (orange).

V. Conclusion and Outlook

With QML-Essentials we provide a framework that aims to give researchers a tool at hand to explore properties of QFMs. To this end, our framework provides algorithms to calculate the expressibility and entangling capability as well as the Fourier coefficients of a provided QFMs. The latter can be achieved by either a FFT or analytical method with the advantage of obtaining the "true" number of coefficients. We implement various tests with the intention of ensuring the accuracy of provided algorithms and aim for a maximum of flexibility to allow adaptation to a variety of use cases.

In future work, we want to extend our framework by adding more testing and validation while also implementing more features including (1) pulse-level control of QFMs, (2) additional measures for entanglement and expressibility, (3) trainable frequencies in the encoding unitaries [5], (4) support for the Qiskit [14] framework and (5) multiprocessing capabilities.

References

[1] M. Ragone *et al.*, "A Lie Algebraic Theory of Barren Plateaus for Deep Parameterized Quantum Circuits," *Nature Communications*, vol. 15, no. 1, p. 7172, Aug. 2024, arXiv:2309.09342 [quant-ph]. DOI: 10.1038/s41467-024-49909-3.

[2] Z. Zimborás *et al.*, *Myths around quantum computation before full fault tolerance: What no-go theorems rule out and what they don't*, arXiv:2501.05694 [quant-ph], Jan. 2025. DOI: 10.48550/arXiv.2501.05694.

[3] M. Schuld, R. Sweke, and J. J. Meyer, "The effect of data encoding on the expressive power of variational quantum machine learning models," *Physical Review A*, vol. 103, no. 3, p. 032430, Mar. 2021, arXiv: 2008.08605. DOI: 10.1103/PhysRevA.103.032430.

[4] M. Wiedmann, M. Periyasamy, and D. D. Scherer, *Fourier Analysis of Variational Quantum Circuits for Supervised Learning*, arXiv:2411.03450, Nov. 2024. DOI: 10.48550/arXiv.2411.03450.

[5] B. Jaderberg *et al.*, "Let Quantum Neural Networks Choose Their Own Frequencies," *Physical Review A*, vol. 109, no. 4, p. 042421, Apr. 2024, arXiv:2309.03279 [quant-ph]. DOI: 10.1103/PhysRevA.109.042421.

[6] N. A. Nemkov, E. O. Kiktenko, and A. K. Fedorov, "Fourier expansion in variational quantum algorithms," *Physical Review A*, vol. 108, no. 3, p. 032406, Sep. 2023, arXiv:2304.03787 [quant-ph]. DOI: 10.1103/PhysRevA.108.032406.

[7] O. Kyriienko and V. E. Elfving, "Generalized quantum circuit differentiation rules," *Physical Review A*, vol. 104, no. 5, p. 052417, Nov. 2021, arXiv:2108.01218 [quant-ph]. DOI: 10.1103/PhysRevA.104.052417.

[8] R. Sweke *et al.*, "Potential and limitations of random Fourier features for dequantizing quantum machine learning," *Quantum*, vol. 9, p. 1640, Feb. 2025, arXiv:2309.11647 [quant-ph]. DOI: 10.22331/q-2025-02-20-1640.

[9] E. Fontana *et al.*, "Spectral analysis for noise diagnostics and filter-based digital error mitigation," arXiv, Tech. Rep., Nov. 2022, arXiv: 2206.08811. DOI: 10.48550/arXiv.2206.08811.

[10] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms," en, *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1 900 070, 2019. DOI: 10.1002/qute.201900070.

[11] W. Mauerer and S. Scherzinger, "1-2-3 reproducibility for quantum software experiments," in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, pp. 1247–1248. DOI: 10.1109/SANER53432.2022.00148.

[12] C. Carbonelli *et al.*, "Challenges for quantum software engineering: An industrial application scenario perspective," in *Quantum Software: Aspects of Theory and System Design*, I. Exman *et al.*, Eds. Cham: Springer Nature Switzerland, 2024, pp. 311–335. DOI: 10.1007/978-3-031-64136-7_12.

[13] V. Bergholm *et al.*, *PennyLane: Automatic differentiation of hybrid quantum-classical computations*, arXiv:1811.04968 [quant-ph], Jul. 2022. DOI: 10.48550/arXiv.1811.04968.

[14] A. Javadi-Abhari *et al.*, *Quantum computing with Qiskit*, arXiv:2405.08810 [quant-ph], Jun. 2024. DOI: 10.48550/arXiv.2405.08810.

[15] D. Seitz *et al.*, "Qadence: A differentiable interface for digital and analog programs," *IEEE Software*, vol. PP, pp. 1–14, Jan. 2025. DOI: 10.1109/MS.2025.3536607.

[16] S. Efthymiou *et al.*, "Qibo: A framework for quantum simulation with hardware acceleration," *Quantum Science and Technology*, vol. 7, no. 1, p. 015018, Jan. 2022, arXiv:2009.01845 [quant-ph]. DOI: 10.1088/2058-9565/ac39f5.

[17] R. Seidel *et al.*, *Qrisp: A Framework for Compilable High-Level Programming of Gate-Based Quantum Computers*, arXiv:2406.14792 [quant-ph], Jun. 2024. DOI: 10.48550/arXiv.2406.14792.

[18] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (Quantum Science and Technology), en. Cham: Springer International Publishing, 2018. DOI: 10.1007/978-3-319-96424-9.

[19] A. Pérez-Salinas *et al.*, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, Feb. 2020, arXiv: 1907.02085. DOI: 10.22331/q-2020-02-06-226.

[20] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951, Publisher: Institute of Mathematical Statistics. DOI: 10.1214/aoms/1177729694.

[21] D. A. Meyer and N. R. Wallach, "Global entanglement in multiparticle systems," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4273–4278, Sep. 2002. DOI: 10.1063/1.1497700.

[22] G. K. Brennen, "An observable measure of entanglement for pure states of multi-qubit systems," arXiv, Tech. Rep., Nov. 2003. DOI: 10.48550/arXiv.quant-ph/0305094.

[23] S. Foulds, V. Kendon, and T. Spiller, "The controlled SWAP test for determining quantum entanglement," *Quantum Science and Technology*, vol. 6, no. 3, p. 035002, Jul. 2021, arXiv:2009.07613 [quant-ph]. DOI: 10.1088/2058-9565/abe458.

[24] T. Haug and M. S. Kim, "Scalable measures of magic resource for quantum computers," *PRX Quantum*, vol. 4, no. 1, p. 010301, Jan. 2023. DOI: 10.1103/PRXQuantum.4.010301.

[25] S. Foulds, O. Prove, and V. Kendon, "Generalising multipartite concentratable entanglement for practical applications: Mixed, qudit, and optical states," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 382, no. 2287, p. 20240411, Dec. 2024, arXiv:2112.04333 [quant-ph]. DOI: 10.1098/rsta.2024.0411.

[26] K. Georgopoulos, C. Emary, and P. Zuliani, "Modeling and simulating the noisy behavior of near-term quantum computers," *Physical Review A*, vol. 104, no. 6, p. 062432, Dec. 2021, Publisher: American Physical Society (APS). DOI: 10.1103/physreva.104.062432.

[27] M. Schuld *et al.*, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, Mar. 2020, arXiv:1804.00633 [quant-ph]. DOI: 10.1103/PhysRevA.101.032308.