# CPEN 400Q Final Project Report

---

Classically Approximating Variational Quantum Machine Learning

with Random Fourier Features

Jerry Xu, Nicholas Ng, Matthew Stefansson, Ying Qi Wen

April 14, 2023

# Table of Contents

# Introduction

In the current quantum computing landscape, many quantum computing applications rely on variational quantum circuits (VQC), especially for working on machine learning (ML) tasks. The VQCs used in these applications are trained similarly to classical machine learning algorithms in order to train the VQCs gate parameters. However, there are a few challenges that arise when using VQCs. One major challenge is the scalability of VQCs. Oftentimes, VQCs require large quantities of quantum resources, especially on large-scale quantum computers, as the number of qubits and/or gates increases.

With this challenge in mind, the paper "Classically Approximating Variational Quantum Machine Learning with Random Fourier Features" by Landman et al. [1] explores the idea of being able to approximate VQCs using Random Fourier Features (RFF) with sampling. This idea stems from it not being "[proven] whether these variational methods can provide a quantum advantage in the general case with a scaling number of qubits" [1]. So, the paper proposes three different strategies of sampling which are Distinct, Tree, and Grid sampling and then applying the RFFs to approximate the VQC. An overview of this process can be seen in figure 1.

The objective of this report is to document all aspects of the work done to replicate the paper by Landman et al. This includes the underlying theory, the results of Landman et al. paper and the replication, the framework design choice, and the details and issues encountered when replicating the paper.

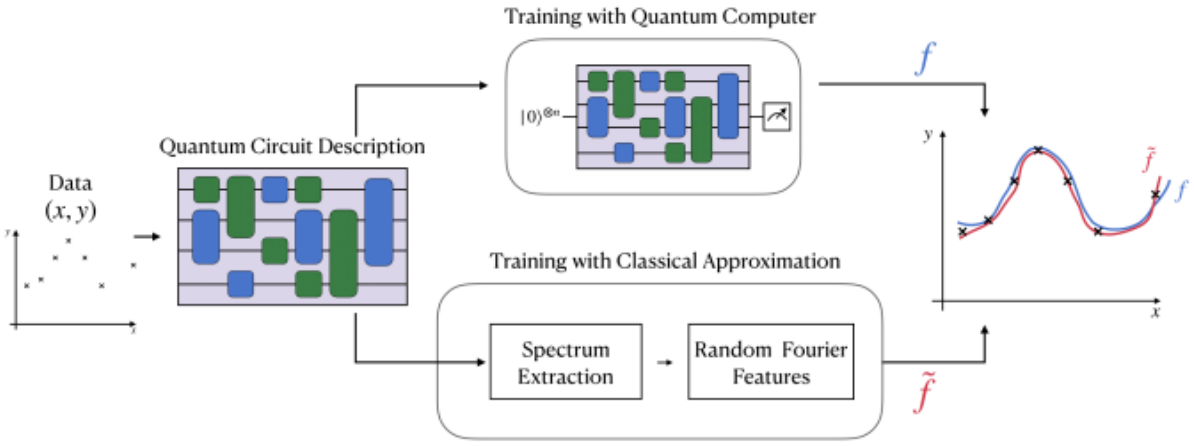



*Figure 1: Diagram explaining the overview of the paper from [1]*

# Theory

The paper is among the more theoretical ones, as it explores the mathematical properties of a VQC, namely its ability to be modelled "well enough" using pure mathematical models.

First, the paper argues that with significant trade off in resources, a VQC can be modelled perfectly by a mathematical model. The paper briefly mentions the work done in [3], which shows that a VQC is mathematically equivalent to a Fourier series whose frequencies can be determined by examining the Hamiltonians in each encoding block. The coefficient vector of the series is then the weight vector to be trained. That is, let $f$ be the function that models the VQC, $\phi$ be the vector consisting of the Fourier series of $f$ without coefficients

$$\tilde{\phi}(x) = \frac{1}{\sqrt{D}} \begin{bmatrix} cos(\omega_i^T x) \\ sin(\omega_i^T x) \end{bmatrix}_{i \in [\![1, D]\!]}$$

we have

$$\tilde{f} = \tilde{\mathbf{w}}^T \tilde{\phi}(x)$$

where $\boldsymbol{w}$ is the weight vector.

Again referring to [3], the paper gives an explicit formula giving all Fourier frequencies. It establishes that the set of frequencies Ω is the set of all possible linear combinations of eigenvalues from distinct Hamiltonians in the encoding blocks. Let $\lambda_i^j$ be the *j*-th eigenvalue of the *i*-th Hamiltonian, define

$$\Lambda_{\boldsymbol{i}} = \lambda_1^{i_1} + \cdots + \lambda_L^{i_L},$$

and

$$\Omega = \left\{ \Lambda_{\boldsymbol{i}} - \Lambda_{\boldsymbol{j}}, \boldsymbol{i}, \boldsymbol{j} \in \prod_{\ell=1}^{L} [|1, d_\ell|] \right\}.$$

Next, the paper proposes that one can construct approximations of f using only select frequencies in the Fourier series. It gives three methods for selecting subsets of, each with their advantages and drawbacks.

## Distinct Sampling

Distinct sampling is the straightforward approach to use RFFs in approximating VQCs. This is a straightforward method of directly sampling from a list of frequencies given by the VQC that is being approximated. The benefit of using this method when compared to the similar method described in [5] is that in [5], the method requires knowing all the frequencies in Ω but in this method we only need to know a subset of the frequencies in Ω. The disadvantage of using this method is that there is no information about how isolated or related the frequencies are and that one may want to sample Ω with probability of occurrence.

In our implementation of distinct sampling, we used the qml.fourier.circuit_spectrum method from the Fourier module. This method allows us to “compute the frequency spectrum of the Fourier representation of simple quantum circuits ignoring classical preprocessing” [4]. By using this it was a simple way to implement the distinct sampling strategy highlighted in the paper.

## Tree Sampling

Tree sampling is a method trying to account for the disadvantage of the distinct sampling by proposing to directly sample from the tree in figure 2. By using this method, it is expected that redundant frequencies are more likely to be sampled than unique frequencies. This is good since the paper’s experiment observes that on average, redundant frequencies are likely to obtain larger coefficients than unique frequencies. However, there is a caveat of using both tree sampling and distinct sampling which is that “if one or more of the encoding Hamiltonians are hard to diagonalize”, then it makes it hard, if not impossible, to sample the frequencies due to missing frequencies, or missing branches for tree sampling.

For our implementation of the project, we only use functions with pauli encoding gates. To do tree sampling, you diagonalize the hamiltonians of the quantum function and then sum the eigenvalues of the tree as shown in figure 2 and 3. After summing the eigenvalues you are left with multiple uppercase lambda sums and to get the frequencies you subtract all the uppercase lambdas to get the set of all frequencies.

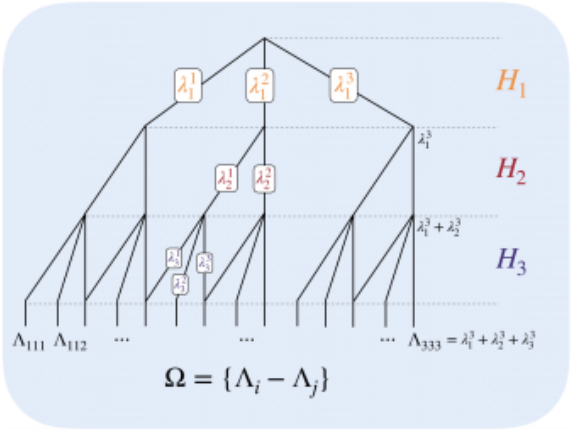



*Figure 2: Tree sampling diagram from hamiltonians to frequencies*

$$\Lambda_i = \lambda_1^{i_1} + \cdots + \lambda_L^{i_L}$$

*Figure 3: Summation equation to calculate uppercase lambda*

## Grid Sampling

Grid sampling is a method that tries to fit the scenario when one or more of the Hamiltonians are hard to diagonalize. This is because grid sampling guesses a range for the frequencies of the quantum circuit from zero to an upper bound that the method guesses. The tradeoff with this method is that if the step size, *s*, of the grid is small then the grid gets very large and complex, especially in higher dimensions.

For our implementation of the project, we only use functions with Pauli encoding gates. To do grid sampling you need to determine a *w_max* value and in our case the *w_max* value is equal to the total number of pauli encoding gates in our quantum function. After determining *w_max* you create an array of frequencies from 0 to *w_max* and you increment the value of each frequency in the array by a value *s* for step. The value s can be any value greater than zero and less than or equal to one. The paper recommends using a step value of 0.1.

## Random Fourier Features Implementation

Appendix A of the paper gives an explicit method for constructing an approximate function using RFF. Given a subset of Ω with *D* frequencies, we first construct the vector $\tilde{\phi}(x)$ given by

$$\tilde{\phi}(x) = \frac{1}{\sqrt{D}} \begin{bmatrix} cos(\omega_i^T x) \\ sin(\omega_i^T x) \end{bmatrix}_{i \in [\![1, D]\!]},$$

and the approximate function $\tilde{f}$ can be computed as follows

$$\tilde{f} = \tilde{\mathbf{w}}^T \tilde{\phi}(x),$$

where $\tilde{\mathbf{w}}'$ is the weight matrix containing trainable parameters. To train $\tilde{\mathbf{w}}'$ is equivalent of finding the following minimum

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{M} \sum_{i=1}^{M} |\mathbf{w}^T \phi(x_i) - y_i|^2 + \lambda ||\mathbf{w}||^2$$

in which λ is a hyperparameter and ***y*** is the vector of the values of the target function at $x_i$

Fortunately, gradient descent can be avoided during training, as the minimization can be computed explicitly as follows:

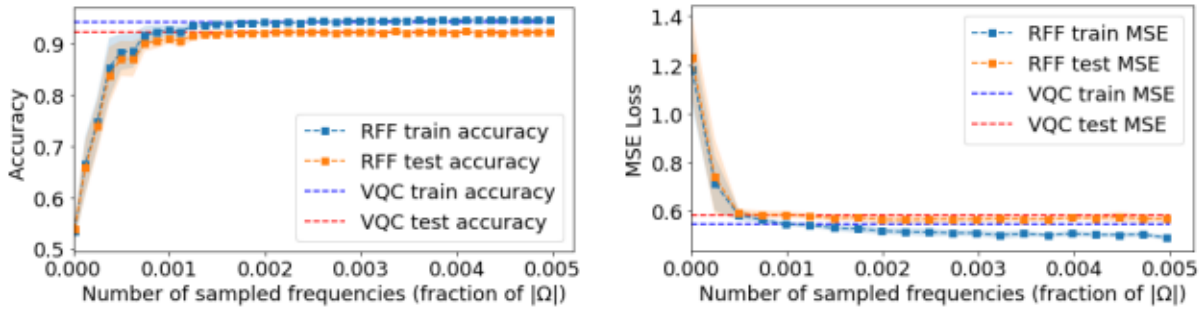
$$\mathbf{w}^* = (\mathbf{\Phi}^T \mathbf{\Phi} + M\lambda I_p)^{-1} \mathbf{\Phi}^T \boldsymbol{y}$$

given that there are more data points than features.

## Results

Getting into the novel results of the paper, Landman et al. study the potential expressive advantage of VQCs for machine learning tasks. The authors of the paper have expressed both theoretical and experimental results. Theoretically, the authors found that the classical sampling method with RFFs was able to approximate VQCs for machine learning tasks. Experimentally, the authors tested the RFF approximators, using many types of datasets, and found that these methods were able to match performance of the VQCs and possibly even surpass them.

Here in figure 4, you can see the performance comparison between the RFF approximator and the VQC. The left graph compares the accuracy between the two methods and they look identical. However, when comparing the mean square error (MSE) in the right graph, it shows the RFF approximator able to slightly outperform the VQC, but only by a marginal amount when looking at the scale.




(a) Fashion-MNIST dataset (classification)

(b) California Housing dataset (regression)

*Figure 4: RFF and VQC train and test accuracy and MSE*

The paper concludes by questioning what quantum advantage means in the context of VQCs because we are able to approximate VQCs using classical computers.

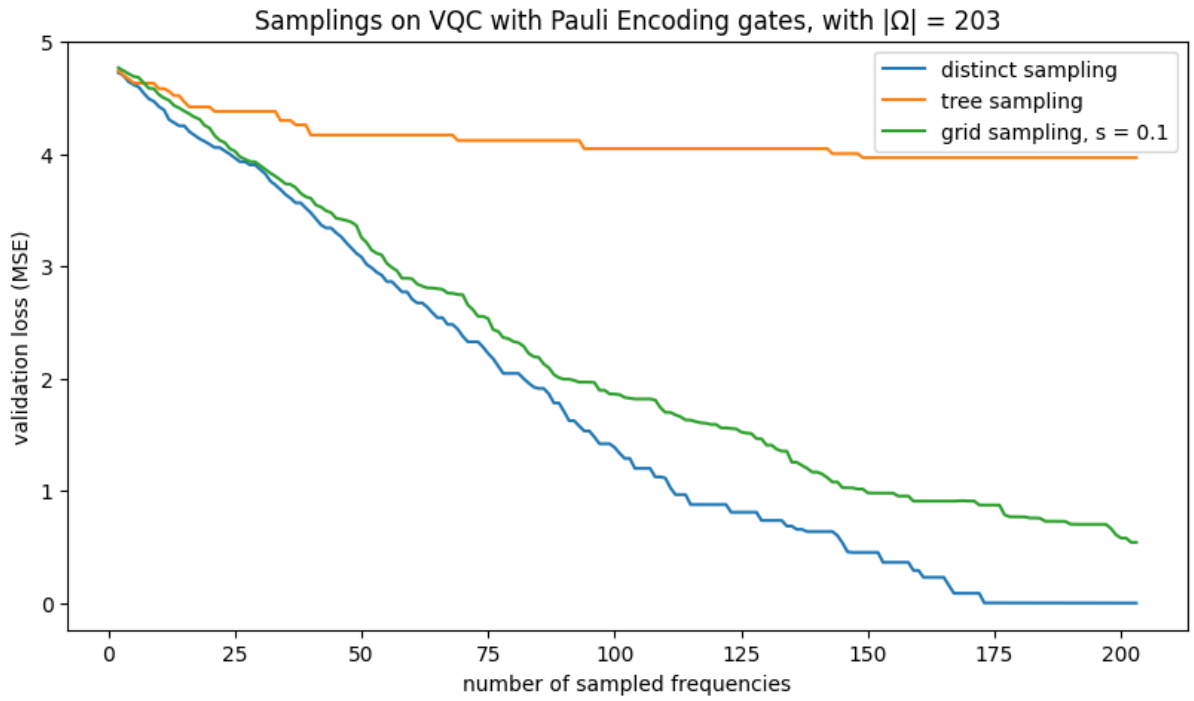
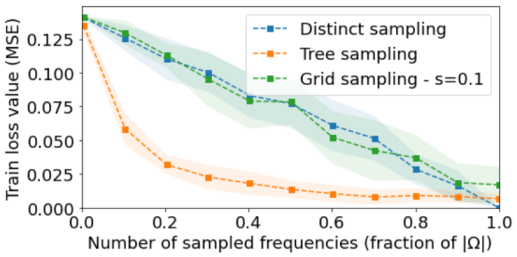


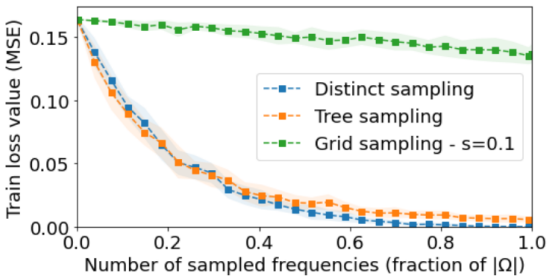

*Figure 5: Our results for the loss of the three sampling methods for a target function with degree 100*

The results of our paper replication can be seen in figure 5. Evaluating this graph, we can see that the distinct and tree sampling methods have similar downward trends of each other, both making their way towards 0. However, in this graph, we can see that tree sampling seems anomalous but also has a downward trend, however the slope is very flat. This could be as a result of sampling from a binomial distribution function. We chose to use this because the regular tree sampling implementation takes too long to run for a binomial tree with 100 layers and 1.26E30 nodes. When comparing our results to the results of the paper, which can be seen in figures 6 and 7, our results match the general trend from the paper for each of the three sampling methods. This is with the caveat that tree sampling shows anomalous results. Our results could also be differing due to using different VQCs or using different datasets.




(b) Evolution of RFF train loss as a function of the relative number of frequencies sampled. The *Tree* sampling strategy takes advantage of the high redundancy to sample less frequencies to reach a good approximation.

*Figure 6: "Random 1d VQCs with L=200 Pauli encoding gates, averaged over 10 different random initialization" [1]*




(b) RFF train loss with different sampling methods on the random VQCs. The Distinct sampling benefit from the concentration of frequencies in packets to approximate with less samples.

*Figure 7: "Random 1d VQC with 4 scaled Paulis and and a 3-qubits HXY Z Hamiltonian" [1]*

# Framework

The Pennylane library for Python was the framework of choice to implement the replication of the paper. There was no particular reason for choosing Pennylane, but after reading the paper thoroughly, there did not seem to be a reason to switch from Pennylane to another framework. This was because the project group deemed Pennylane to be sufficient in being able to satisfy our software needs in the beginning. Additionally, this was the case because there were no intensive machine learning algorithms to implement in our paper since it focuses on approximating VQCs using a classical method.

The Fourier module in Pennylane was used in the first sampling algorithm, distinct sampling, because, as suggested by the professor, it is able "to analyze the Fourier representation of quantum circuits" [2]. Namely, the circuit_spectrum function, if given a quantum circuit, outputs the spectrum of frequencies for that circuit. However, we decided not to use the

function and implemented the latter two sampling methods by calculating the spectrum following the instructions and formulas outlined in the paper. In this way, we gained a deeper understanding of how the frequencies are calculated and also more flexibility when sampling the frequencies. For example, we could define our own upper bound frequency and the step size when working with grid sampling, which would be otherwise impossible if using the circuit_specturm function.

After beginning implementation and also by the midpoint, there was no reason to switch frameworks. No issues in terms of software implementation up until this point were encountered, such as if a machine learning algorithm took too long to run, or if it was unfeasible to implement the different sampling strategies with RFFs.

# Reproducibility

The paper by Landman et al. seemed to be a relatively easy paper to replicate compared to some of the papers presented in class. The major issue when replicating this paper was the notation issues within the paper. One big issue that can be found in the paper is seen in figure 8 and 9 which is the definition of ϕ and tilde ϕ. This is because in the paper describes tilde ϕ having length of 2*D*, which is two times the number of sampled frequencies, and describes ϕ having length *p*, which is the number of features and *p* is equivalent to *D*.

$$\tilde{\phi}(x) = \frac{1}{\sqrt{D}} \begin{bmatrix} cos(\omega_i^T x) \\ sin(\omega_i^T x) \end{bmatrix}_{i \in [\![1, D]\!]}$$

*Figure 8: Equation to calculate tilde* ϕ

where $\mathbf{\Phi}$ is a matrix of size $M \times p$ with each row $i$ corresponds to $\phi(x_i)^T$ and $\boldsymbol{y}$ is the vector of all the labels $y_i$.

*Figure 9: Excerpt from paper describing the size of* ϕ

To try and resolve this issue, the project group tried to reach out to both the leading author and a supporting author, which the professor knew, but did not receive a response until the morning of the presentation day. Therefore, we had to work based on assumptions, especially in the work to construct the kernel and solve the linear regression problem. The assumption made to resolve the notation issues in the paper was to assume that $p = 2D$ and from the results, it seemed to have been a satisfactory resolution.

One assumption made to simplify the process was disregarding the calculation for *D* found in figure 10. This is due to the circuits used in the experiment being simple and giving one or two digit numbers of frequencies. Instead of sampling a subset of, for example, seven frequencies, sampling all the frequencies was reasonable given the simple circuits. Comparing this to the work of Landman et al., the authors were working with a number of frequencies many magnitudes larger, reaching up to about $10^7$ than the replication, reaching about $10^1$.

$$D = \Omega\left(\frac{dC_1(1+\lambda)^2}{\lambda^4\epsilon^2}\left[log(dL^2|\mathcal{X}|)+log\frac{C_2(1+\lambda)}{\epsilon\lambda^2}-log\delta\right]\right) \tag{22}$$

*Figure 10: Equation for number of frequencies to calculate*

# Future work

Although we had a great start for this project and we gained a deep understanding of the theory behind the paper, the work is not over yet. For the future, there are a couple of things that we can improve or implement. Firstly, we would do research and try to find a much more complex VQC with many more encoding gates. In this way, we would obtain a wider range of frequencies, allowing us to properly sample them and see the different effects between different sampling strategies. Secondly, we want to test the procedure on a real dataset. The author compared VQC and RFF approximation on Fashion-MNIST and California Housing datasets. We had the intention to do the same, however, it would result in a significant amount of work which should be approached carefully instead of rushing through. In addition, it is likely that this step requires a more complex VQC than the one we had. So we would have many prerequisites to finish first.

# Appendix: References


1. https://arxiv.org/abs/2210.13200v1
2. https://docs.pennylane.ai/en/stable/code/qml_fourier.html
3. https://arxiv.org/abs/2008.08605
4. https://docs.pennylane.ai/en/stable/code/api/pennylane.fourier.circuit_spectrum.html?highlight=circuit_spectrum#pennylane.fourier.circuit_spectrum
5. https://arxiv.org/abs/2206.11740