Path Integral Based Quantum Neural Network

Uddhav Sen

1 INTRODUCTION

The entire theory of non-interacting quantum mechanics can be reformulated from Feynman's path integral formulation. The Feynman Kernel exhibits remarkable symmetry and can construct not only the traditional first quantisation quantum mechanics but also extends to more exotic quantum field theoretic processes explained by Feynman diagrams. The Schrodinger picture can be viewed as a special case of the more general Feynman process. On the other hand, recent advances in building scalable quantum computers has led researchers to search algorithms inspired by quantum mechanical laws which can be efficiently implemented and are outperforming the best performing classical ones. There exists quite a few remarkable results including the quantum search, factoring, superdense coding to name a few. The entire scope of **BQP**(Bounded Error Quantum Polynomial time) complexity class which ascertains quantum supremacy over the best performing classical algorithms is still not completely understood. Meanwhile in computer science, statistical learning theory is a rapidly emerging field since it's inception by Vapnik et. al. and it has evolved into the well known branches of supervised and unsupervised learning, pattern recognition, evolutionary systems etc. Recently, there has been a spark of interest in the subject by the physics community as the similarities between artificial neural networks and statistical physics has gotten clearer. The current focus from a quantum information perspective has been identifying modes of Information flow in traditional neural networks and extending similar ideas to construct quantum neural networks which has led to ideas like hybrid quantum neural network models like TensorFlow Quantum and others in an attempt to harness optimal information processing capabilities in a quantum computer. The primary focus is to utilise quantum data and identify patterns in such data using a hybrid mixture of classical and quantum algorithms.

Despite having significant success in constructing intermediate scale quantum systems of the order of 100 qubits and developing integrated frameworks to simulate hybrid learning algorithms in an effort to achieve 'Quantum Supremacy'; there is a lack of a unified mathematical structure which correctly represents the theory of quantum machine learning. Current research focuses on the extension of classical neural network architectures to parametric quantum circuits and establishing a quantum loss function to deriving optimal parameter training results from a quantum perspective. A very successful attempt in the aforemen-

tioned direction is the Variational Quantum Algorithm(VQA) technique which trains a set of quantum gates by tweaking parameters using a classical optimiser. The central set of objectives of the paper is to show how the Feynman path integral formulation(abbreviated as FPI from here) is a candidate algorithm for simulating highly correlated quantum data as the discrete Feynman Kernel in computational Hilbert Space naturally orients itself into a quantum generalisation of a neural network.

2 Objectives Of The Paper

The paper aims to establish the central idea of the Feynman Kernel as the core mathematical structure which explains the learning architecture or the optimal evolution of a quantum neural network. This is established in the given sequence as:

- The FPI formulation is shown as an alternative model for creating traditional quantum computing algorithms which are generally built from quantum circuits. The equivalence can be established from the discretised kernel which is used to derive a revealing geometrical picture of evolution in finite dimensional abstract Hilbert Space and serves as an alternative viewpoint from the Bloch picture. The geometrical picture further leads to ideas of how Information might be processed in the kernel.
- It is shown how the discretised kernel decomposes into a neural net architecture in the following section. We also show how to represent uncertainty by decomposing the Feynman neural net into a Fourier transform to optimally select the most informative and hence unresolved features on classical datasets. A measure of randomness is chosen which helps to quantify the suitability of the potential chosen for system activation.
- The last objective is to prove the efficiency of the model over other architectures and a general algorithm to efficiently simulate such a system. We also provide a future outlook into the possibilities of extension into pure quantum learning theory.

3 Connection between the Feynman Kernel and the Quantum Circuit Model

The wave function after state evolution in the FPI formulation is given as $\psi(\mathbf{x},t) = \int \mathcal{D}\mathbf{x}(t) \exp\Big(i\frac{S[\mathbf{x},\dot{\mathbf{x}}]}{\hbar}\Big)\psi(\mathbf{x},0)$ where $\mathcal{D}\mathbf{x}$ represents the functional path differential in infinite dimensional Hilbert Space and the rest have their usual meaning. It is sufficient for the path differential to convert as an Euclidean product since we are primarily interested in computational Hilbert Space which is finite dimensional and consequently the wave function becomes $\psi(x_N = b, t) =$

$\prod \int_{x_j} dx_j \exp\Big(i\frac{S[\mathbf{x},\dot{\mathbf{x}}]}{\hbar}\Big)\psi(x_0,0) = \frac{1}{Z}\int_{x_0}..\int_{x_{N-1}} \prod_{j=1}^{N-1} dx_j \exp\Big(i\frac{S[x_j,\dot{x_j}]}{\hbar}\Big)\psi(x_0,0)$
where Z is the normalisation factor. The transition amplitude or the Feynman kernel is explicitly given by $K(b;a) = \frac{1}{Z}\int_{x_1}..\int_{x_{N-1}} \prod_{j=1}^{N-1} dx_j \exp\Big(i\frac{S[x_j,\dot{x_j}]}{\hbar}\Big)$ where $x_0 = a$ and $x_N = b$ is the starting and ending point respectively. The derivation of equivalence in the Hamiltonian simulation and the FPI kernel is given in []. This leads to the idea of visualising the traditional quantum circuit model alternatively from a different perspective; viz. using FPI.

3.1 Mathematical construction of the Discretised Feynman Kernel in computational Hilbert Space

The Feynman kernel described above can be also defined from events occurring in succession as:

$$K(b;a) = \frac{1}{Z}\int_{x_1} .... \int_{x_{N-1}} \prod_{j=1}^{N-1} K(j+1,j)\ dx_j$$

where $K(j+1,j) = exp(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{x_{j+1}+x_j}{2}, \frac{\dot{x}_{j+1}-\dot{x}_j}{\epsilon}, \epsilon))$ is the corresponding kernel for the x_j point. This is same as the one defined before but calculations are easily carried out in the latter. The process begins by the creation of a lower bound Reimann sum as:

$$\begin{aligned}\int_{x_j} K(j+1,j)dx_j &= \int_{x_j} exp(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{x_{j+1}+x_j}{2}, \frac{x_{j+1}-x_j}{\epsilon}, \epsilon))dx_j \\ &\approx \sum_{n_j=-L}^{n_j=L} exp(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{n_{j+1}+n_j}{2}, \frac{n_{j+1}-n_j}{\epsilon}, \epsilon))\delta x_j \qquad (1)\\ &= f(n_{j+1})\end{aligned}$$

The discrete kernel for a single step is represented by (1) where the continuous position operators are replaced by (n_j, n_{j+1}) pairs respectively and it is evident that it is a function of n_{j+1}. Similarly for $j+1$;

$$\begin{aligned}&\int_{x_{j+1}} \Big(\int_{x_j} K(j+1,j)dx_j\Big)\Big(K(j+2,j+1)\Big)dx_{j+1} \\ &\approx \lim_{\delta x_{j+1}\to 0, N\to\infty} \sum_{n_{j+1}=-L}^{L} f(n_{j+1}) exp(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{n_{j+2}+n_{j+1}}{2}, \frac{n_{j+2}-n_{j+1}}{\epsilon}, \epsilon))\delta x_{j+1} \\ &= \lim_{\delta x_{j+1}\to 0, L\to\infty} \sum_{n_{j+1}=-L}^{L} \Big[\sum_{n_j=-L}^{L} exp(\frac{i\epsilon}{\hbar}(\mathcal{L}_j + \mathcal{L}_{j+1}))\Big]\delta x_j \delta x_{j+1} \qquad (2)\end{aligned}$$

The partition size is taken to be equidistant $\delta x_j = c\ \forall j$ and since the kernel will be normalised; they can be ignored without any effect on the kernel. Equation

(2) reveals the structure of the discrete transition amplitude after $N-1$ steps to be

$$K^*(b;a) \sim \sum_{\substack{\{n_j\}\in[-L,L] \\ \forall 1\leq j\leq N-1}} \prod_{j=1}^{N-1} exp\Big(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{n_{j+1}+n_j}{2}, \frac{n_{j+1}-n_j}{\epsilon}, \epsilon)\Big)$$

$$= \frac{1}{Z} \sum_{\substack{\{n_j\}\in[-L,L] \\ \forall 1\leq j\leq N-1}} \prod_{j=1}^{N-1} exp\Big(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{n_{j+1}+n_j}{2}, \frac{n_{j+1}-n_j}{\epsilon}, \epsilon)\Big) \quad (3)$$

Here Z is the normalisation factor and there is negligible deviation of $K^*(b;a)$ from the continuum limit. The wavefunction thus formed at the final point b is calculated by varying $K^*(b,a)$ by varying the initial point a along with the intermediate events as;

$$\langle b|\psi_f\rangle = \frac{1}{Z}\sum_{j=0}^{N-1} K^*(b;a_j)\lambda_j \quad (4)$$

where the initial wavefunction at a is given by $|\psi_0\rangle = \sum_{j=0}^{N-1}\lambda_j\,|j\rangle$. Similarly the final wavefunction can be derived from (4) by taking all possible final destination b.

3.2 Geometrical evolution of $K^*(b,a)$

The evolution of quantum circuits can be effectively visualised as rotations on the Bloch Sphere which aids in geometrical understanding of the physical system. A similar representation also needs to be created in the FPI model of quantum computation to visualise the complex formulas mentioned above. The points are the possible positions the particle can occupy at a given time instant and the separation is characterised by the width ϵ. The total number of layers of computation is equal to the number of points/possibilities represented by eigenkets in Hilbert Space. The traditional computational Hilbert Space marked by $\{|j\rangle\}$ is the corresponding set of basis vectors for the given set of finite points. The entire traditional Feynman picture in continuous spectrum can be reconstructed from the given diagram which enforces the idea behind (3). The diagrams explain the geometrical evolution of the system after the introduction of computational basis. The next diagram illustrates the scenario when the final point is a set of possibilities; i.e. the final state is unknown(for 2 and 2^n level systems). The algorithm for the entire quantum simulation is given by:

Diagrams 1 and 2 serve as the geometrical model of the path integral method of quantum computing just as the Bloch sphere represents the quantum circuit model inspired by Hamiltonian simulations. They also suggest a possible hidden information processing system being exhibited by the Kernel simulation

Algorithm 1: Quantum Simulation Algorithm in FPI Model

Result: The Probability amplitude of the particle/qubit in a chosen basis $|b\rangle$

Input : A set of computational basis $\{|j\rangle\}$

Input : A fixed time step ϵ

Input : Initial wave function of the qubit $|\psi_0\rangle = \sum_{j=0}^{N-1} \lambda_j |j\rangle$

Input : Potential Setup for each layer of computation $\{V_j\}$

1 Set $K^*(j+1, j) = exp(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{x_{j+1}+x_j}{2}, \frac{\dot{x}_{j+1}-\dot{x}_j}{\epsilon}, \epsilon))$

2 $\forall j \in [1, N-1]$, multiply every Kernel such that after N-1 steps it becomes $\Pi_{j=1}^{N-1} K^*(j+1, j)$

3 $\forall j \in [1, N-1]$, sum through all possible points j to get $K^*(b; a) = \sum .... \sum K^*(j+1; j)$

4 The probability amplitude is $\langle b|\psi_f\rangle = \frac{1}{Z}\sum \lambda_j K^*(b; a_j)$

approach if the events marked as space lattices are marked as nodes in abstract computational space. Each computational layer can be represented as a single layer of a hidden layer in a deep neural network. The rigorous verification of such possibilities are extensively explored in the next section.

4 Kernel decomposition into a neural network

Claim 1: In a classical neural network, the goal is to minimise a cost function $C(\vec{\theta})$ or equivalently find an optimal connection matrix W s.t. $||(y - Wx)|| \to 0$ where x, y are the input and target datasets respectively. We show that it is possible to generalize a fully connected classical NN to a quantum neural net structure by replacing x, y vectors with initial, first and final activation layer $|\psi_0\rangle, |\psi_1\rangle$ and $|\psi_2\rangle$ such that every node is a basis in a finite Hilbert Space, and by constructing the connection matrix using the discrete Feynman path integral kernel. The derivation of the cost function is delayed till the next section.

The path integral decomposes into the partition function after setting $t = -\iota\beta$ as shown in [cite] which gives further insight into how $K^*(;)$ can be used to efficiently construct a quantum neural network. We have already shown how to simulate quantum algorithms by selecting a suitable Lagrangian in the kernel approach. The immediate proof of $K^*(;)$ simulating the quantum activation function is shown for a two level quantum system i.e. $\{|0\rangle, |1\rangle\}$. The diagram shows three computational layers linked with Lagrangian action. The Kernel $K^*(;)$ is generated from a feed-forward hidden neural network as indicated by the diagram. The wave-function after an ϵ time step by the discrete Feynman

process is;

$$a'_0 = exp\Big(\frac{iS^1_{00}}{\hbar}\Big) + exp\Big(\frac{iS^1_{10}}{\hbar}\Big) \tag{5}$$

$$a'_1 = exp\Big(\frac{iS^1_{01}}{\hbar}\Big) + exp\Big(\frac{iS^1_{11}}{\hbar}\Big) \tag{6}$$

After the next ϵ time step $|\psi_1\rangle \rightarrow |\psi_2\rangle$ with corresponding amplitudes;

$$\begin{aligned} a"_0 = exp\Big(\frac{iS^1_{00}}{\hbar}\Big)exp\Big(\frac{iS^2_{00}}{\hbar}\Big) + exp\Big(\frac{iS^1_{10}}{\hbar}\Big)exp\Big(\frac{iS^2_{00}}{\hbar}\Big) + \\ exp\Big(\frac{iS^1_{11}}{\hbar}\Big)exp\Big(\frac{iS^2_{10}}{\hbar}\Big) + exp\Big(\frac{iS^1_{01}}{\hbar}\Big)exp\Big(\frac{iS^2_{10}}{\hbar}\Big) \end{aligned} \tag{7}$$

$$\begin{aligned} a"_1 = exp\Big(\frac{iS^1_{00}}{\hbar}\Big)exp\Big(\frac{iS^2_{01}}{\hbar}\Big) + exp\Big(\frac{iS^1_{01}}{\hbar}\Big)exp\Big(\frac{iS^2_{11}}{\hbar}\Big) + \\ exp\Big(\frac{iS^1_{10}}{\hbar}\Big)exp\Big(\frac{iS^2_{01}}{\hbar}\Big) + exp\Big(\frac{iS^1_{11}}{\hbar}\Big)exp\Big(\frac{iS^2_{11}}{\hbar}\Big) \end{aligned} \tag{8}$$

which can be re written as;

$$a"_0 = exp\Big(\frac{iS^2_{00}}{\hbar}\Big)a'_0 + exp\Big(\frac{iS^2_{10}}{\hbar}\Big)a'_1 \tag{9}$$

$$a"_1 = exp\Big(\frac{iS^2_{01}}{\hbar}\Big)a'_0 + exp\Big(\frac{iS^2_{11}}{\hbar}\Big)a'_1 \tag{10}$$

The structure represents a linear relationship between the amplitudes with Lagrangian path connections which resembles a feed forward neural network with activation layer given by $y = Wx$ where W is the weight matrix. This leads to the idea of the discretised Feynman process as a possible mathematical structure of being capable of simulating quantum neural networks.

4.1 N-Dimensional quantum neural net construction

The possible existence of a Feynman path integral inspired quantum neural network(abbreviated as PINN) is validated by considering a general $N = 2^n$ dimensional wave-function in computational Hilbert Space. We consider the initial wave-function as $|\psi_0\rangle = \sum_{j=0}^{2^n-1} a_j |j\rangle$ and assume a computational cycle of $N = 2^n - 1$ computations for each time step ϵ as stated in the previous section. We fix a basis $|l"\rangle$ in the second time step and calculate the total contribution. The corresponding points in the second and third layer layer is differentiated by ', " from the first layer as similar to the previous section. The total contribution to the amplitude at $|l"\rangle$ in the second cycle keeping the first step fixed at $|0'\rangle$ as indicated by the diagram is;

$$\begin{aligned} K^*(l",)_{0'} &= K^*_1(0',0)K^*_2(l",0') + K^*_1(0',1)K^*_2(l",0')....+ K^*_1(0',2^n-1) \\ &\quad K^*_2(l",0') \\ &= K^*_2(l",0')\sum_{j=0}^{2^n-1} K^*_1(0',j) \end{aligned} \tag{11}$$

Similarly, the total contribution to $|l"\rangle$ from $|1'\rangle$ is;

$$\begin{aligned} K^*(l",)_{1'} &= K_1^*(1',0)K_2^*(l",1') + K_1^*(1',1)K_2^*(l",1').... + K_1^*(1',2^n-1) \\ & K_2^*(l",1') \\ &= K_2^*(l",1')\sum_{j=0}^{2^n-1} K_1^*(1',j) \end{aligned} \tag{12}$$

The total contribution at $|l"\rangle$ after varying the entire second layer becomes

$$K^*(l",) = \sum_{j,j'} K_2^*(l",j')K_1^*(j',j) \tag{13}$$

We also know that $b_{j'} = \sum_j a_j exp\Big(\frac{iS^1_{j',j}}{\hbar}\Big)$ from the discrete Feynman process. After substituting $K_1^*(j',j) = exp\Big(\frac{iS^1_{j',j}}{\hbar}\Big)$ and $K_2^*(l,j') = exp\Big(\frac{iS^2_{l",j'}}{\hbar}\Big)$ in (13); we get

$$K^*(l",) = \sum_{j,j'} exp\Big(\frac{iS^1_{j',j}}{\hbar}\Big)exp\Big(\frac{iS^2_{l",j'}}{\hbar}\Big) = \sum_{j,j'} exp\Big(\frac{i(S^1_{j',j} + S^2_{l",j'})}{\hbar}\Big) \tag{14}$$

The initial distribution is given by $|\psi_0\rangle$ and hence the probability amplitude of $|l"\rangle$ is

$$c_{l"} = \sum_{j,j'} a_j exp\Big(\frac{i(S^1_{j',j} + S^2_{l",j'})}{\hbar}\Big) \tag{15}$$

This process can be continued for $(2^n - 1)$ computational cycles and we would get the final wave vector as the simulation algorithm. The above mentioned procedure is a detailed work-out for the first two ϵ cycles. An interesting observation to note is that the node connections are linear and follows $y = W_1W_2x$ where the Weight matrices are given by the path connections in the system. This implies that the discrete Kernel naturally decomposes into a quantum neural network. The mathematical structure of the Kernel is very suitable for various QNN construction and a finalised diagram is shown.

4.2 Decomposition into a Linear Neural Network

A major difference between the implementation of a quantum simulation using the Feynman Kernel and a PINN model is the selection of potential setups for consecutive layers of computation. There always exists a fixed potential function chosen for a quantum simulation algorithm whereas we aim to optimally select a family of potentials for optimal stochastic evolution in a PINN setup. The basic idea is to optimally process information in a stochastic quantum data-set with minimal noise. The current section does not dive deeper into the process of the quantum information dynamics of a PINN which would ultimately drive

the optimal potential selection for a specific problem. For real physical system simulation the Lagrangian will be following similar patterns but for abstract processes in quantum computing; $\mathcal{L} = \alpha f_1 + \beta f_2$ is chosen where $\alpha, \beta \in \mathbf{R}$ and f_1, f_2 are suitably chosen functions in the finite space. The kernel has a general form of $K^*(b,a) \sim \sum .... \sum \prod K^*(i+1,i)$ where $K^*(i+1,i) = exp(\frac{i\epsilon\mathcal{L}}{\hbar})$. The decomposition of the original PINN into a linear evolution occurs only when $S < \hbar$ which resembles the traditional architecture and is shown in the Appendix.

4.3 PINN as a Natural Information Processing Model

We have shown how to explicitly perform quantum simulation using the Feynman kernel in computational basis as an alternative to gate based quantum computing and subsequently shown how the structure can possibly perform quantum machine learning. This section lays out the structure to establish a connection between optimal quantum information processing and a natural process furnished by the Feynman Kernel. We start with $|\psi_0\rangle = \sum_k a_k |k\rangle$ and we evaluate the Weight matrix due to the PINN as

$$W = \begin{bmatrix} exp(i\theta_{0,0}) & exp(i\theta_{1,0}) & . & . & exp(i\theta_{N-1,0}) \\ exp(i\theta_{0,1}) & exp(i\theta_{1,1}) & . & . & exp(i\theta_{N-1,1}) \\ . & . & . & . & . \\ . & . & . & . & . \\ exp(i\theta_{0,N-1}) & exp(i\theta_{1,N-1}) & . & . & exp(i\theta_{N-1,N-1}) \end{bmatrix} \tag{16}$$

Where $\theta_{\alpha,\beta} = \frac{\epsilon\mathcal{L}_{\alpha,\beta}}{\hbar}$ and $\mathcal{L}_{\alpha,\beta} = \frac{1}{2}m\left(\frac{x_\beta - x_\alpha}{\epsilon}\right)^2 - V(x_\alpha, x_\beta)$ where β is the target basis node and α is the starting basis node.

We aim to establish the following set of ideas in the next section;

1. We show how the PINN architecture formally defined in Section 5.1 and geometrically represented in Section 5.2 requires to optimise the same cost function structure as required by the VQA model to be categorised as a QNN.

2. We prove how the FPI Kernel serves as a theoretical limit to QNN training. This upper limit signifies optimal Information processing in a QNN architecture.

3. The condition for relative optimality of a PINN inspired operator over another is investigated.

5 QAQD

5.1 General Quantum Optimisation problem formulation

Claim 2.1: We show an equivalent picture of PINN formulation using the density operator instead of wave-vectors.

Statement 2.2: We formally introduce the structure of PINN inspired QNN to facilitate parametric quantum information processing with the help of a diagram.(Write the equivalence of density picture in description)

Claim 2.3: We claim how the PINN inspired kernel serves as an upper theoretical limit to the total amount of Information present in the system. A cost function with added temporal dependence is defined which captures the inherent trade-off between generalisation capabilities and accuracy/convergence rate.

As we have shown that the Feynman path integral model can be visualised as a quantum neural network; the first objective is to formulate a generic quantum learning problem and look for a possible connection with current existing quantum machine learning models in use. The W matrix characterised as $W_{\alpha,\beta} = exp(i\theta_{\alpha,\beta})$ marks the Feynman inspired evolution in a non-unitary gate based model where $\theta_{\alpha,\beta} = \frac{\epsilon \mathcal{L}_{\alpha,\beta}}{\hbar}$ for pure physical simulations and $\theta_{\alpha,\beta}$ is monotonically variant on ϵ for all cases. The physical structure of the PINN matrix allows a single parameter(say θ_m) to describe an entire phase space where it takes the path connections as input $(\alpha, \beta) \in [0, 2^n - 1]$ only. The evolution of an initial state vector $|\psi_0\rangle$ after l cycles of parametric evolution under PINN would look like $|\psi_l\rangle = \prod_{m=1}^{l} W_m(\theta_m) |\psi_0\rangle$. Obviously $|\psi_l(\theta_1, .., \theta_l)\rangle$ is a $l-$ dimensional parametric wavefunction. A more complete picture can be obtained by using the density operator formalism as it allows mixed states and we shall fall back to the wave-function picture for a more vivid geometric representation whenever possible.
The same idea of PINN inspired evolution for an initial state $\rho_0 = \sum_k p(k) |\psi_k\rangle \langle\psi_k|$ is given as;

$$\rho_f(\theta_1, .., \theta_l) \sim W_l(\theta_l)...W_1(\theta_1)\rho_0 W_1^+(\theta_1)...W_l^+(\theta_l) \tag{17}$$

The system obviously blows up since W_m is not norm preserving. Going back to the wave description; for non unitary quantum evolution we define post evolution as;

$$|\psi_1(\theta_1)\rangle = \frac{W_1(\theta_1) |\psi_0\rangle}{\langle\psi_0| W_1^+(\theta_1) W_1(\theta_1) |\psi_0\rangle} \tag{18}$$

The same process of normalisation is followed after every implementation of the $\{W_m(\theta_m)\}$ operators which is known as the post measurement state. The general process of circuit implementation of non unitary quantum gates as measurement operators is discussed later. Our main focus from hereon will be on (17) and how to construct a general quantum learning problem assuming that a universal construction is possible as mentioned in[].

The general idea is to optimise the probability of measurement in a selected basis; say $|\zeta\rangle$. The total probability of getting ζ as a measurement outcome is given by;

$$p(\zeta, \vec{\theta}) = \sum_k p(k)p(\zeta|k) = tr\Big(W_\zeta^+(\vec{\theta}) W_\zeta(\vec{\theta}) \rho_0\Big) = tr\Big(W_\zeta(\vec{\theta}) \rho_0 W_\zeta^+(\vec{\theta})\Big) \tag{19}$$

Here $W_\zeta(\vec{\theta}) = \prod_{m=1}^{l} W_m(\theta_m)$ is the complete PINN evolution implemented as an operator with projective measurement outcome as ζ and ρ_0 is the initial density matrix. It should be noted that $|\zeta\rangle$ is one of the basis states and extremising $p(\zeta, \vec{\theta})$ is the objective of a quantum learning problem. The definition of (19) is the traditional VQA structure of $C(\vec{\theta}) = \sum_k f_k(tr(O_k U(\vec{\theta})\rho_0)U^+(\vec{\theta}))$ where the operators are unitary. The primary difference in the PINN based QNN with the VQA is with the evolution architecture or the Ansatze as presented in the paper. The entire process of parametric quantum information processing facilitated by PINN has $\vec{\theta}$ and the individual training time parameters $\vec{\epsilon}$ as input to the model.
The Feynman Kernel $K(b;a)$ in the continuum limit is the upper theoretical limit on variable training because it imitates natural evolution. The process captures the total Information pool required to describe the final state and the closest approximation to the natural path history architecture in computational Hilbert Space it is given by (3). We replace physical action $\mathcal{L}$ with ω_m such that $\epsilon\omega_m = \theta_m$ and $\omega_m = \frac{\mathcal{L}}{\hbar}$ for physical problems. We aim to find a general set of connection matrices to describe complete evolution as;

$$\begin{aligned} K^*(b;a) &\sim \sum_{\substack{\{n_j\}\in[-L,L] \\ \forall 1\le j\le N-1}} \prod_{j=1}^{N-1} exp\Big(\frac{i\epsilon}{\hbar}\mathcal{L}(\frac{n_{j+1}+n_j}{2}, \frac{n_{j+1}-n_j}{\epsilon}, \epsilon)\Big) \\ &\sim \sum_{\substack{\{n_j\}\in[-L,L] \\ \forall 1\le j\le N-1}} \prod_{j=1}^{N-1} exp\Big(\frac{i\epsilon}{\hbar}\mathcal{L}(j, j+1)\Big) \\ &\sim \sum_{\substack{\{n_j\}\in[-L,L] \\ \forall 1\le j\le N-1}} \prod_{j=1}^{N-1} exp\Big(i\epsilon\omega_m(j, j+1)\Big) \end{aligned} \tag{20}$$

The components of the final wave-vector is given by (4). We have already shown how the kernel decomposes into linear operators which leads (20) to become;

$$\begin{aligned} |\psi_f(\omega_m)\rangle = \sum_j K^*(b;j)\,\langle j|\psi_0\rangle\,|j\rangle &\sim \sum_{\substack{\{n_j\}\in[-L,L] \\ \forall 1\le j\le N}} \prod_{j=1}^{N-1} exp\Big(i\epsilon\omega_m(j, j+1)\Big)\lambda_j\,|j\rangle \\ &\sim W^N(\omega_m, \epsilon)\,|\psi_0\rangle \end{aligned} \tag{21}$$

(21) is true $\forall m$ and it is obvious that N is the number of independent bases in Hilbert Space. The amount of Information learnt about ω_m is represented by the Information trajectory paths marked by the connections and for a N layered PINN model as shown in the diagram. Since we know that this natural process furnishes a QNN; $|\psi_f(\omega_m)\rangle$ is the optimal amount of learning required. For two computational cycle intervals $\epsilon > \epsilon' > 0$; we get two PINN evolution $W = W^N(\omega_m, \epsilon)$ and $W' = W^N(\omega_m, \epsilon')$ respectively such that the total evolution period in ω_m becomes $t = N\epsilon$ and $t' = N\epsilon'$ respectively. We can define an optimal

spacing ϵ^* for a specific QNN problem such that for a chosen basis $|\zeta\rangle$; the computational transition probability amplitude $\langle\zeta| W^N(\omega_m, \epsilon^*) |\psi_0\rangle$ is an extremum. As we know that $W(\epsilon^*, \omega_m)$ is implemented as a measurement operator in the circuit model; the amplitude is equivalently $\langle\psi_0| (W_\zeta^+)^N(\omega_m, \epsilon^*) W_\zeta^N(\omega_m, \epsilon^*) |\psi_0\rangle$ an extremum. While considering a mixed state; this amplitude becomes similar to (19) which resembles the cost function. This shows how we can also optimise the time interval between cycles as a parameter in PINN. The inclusion of space-time complexity in (19) suggests that for $\vec{t} = N\vec{\epsilon}$; $C(\vec{\omega}, \vec{t}) = \sum_k f_k(tr(O_k(W_\zeta^+)^N W_\zeta^N \rho_0))$ is the global cost function such that W_ζ is dependent on the entire variable space. For local ω_m, ϵ_m; we can also construct a local cost function if needed similarly. The inclusion of evolution time as a variable leads to an interesting conclusion. Lesser ϵ' implies more detailed Information trajectories covering ω_m space for W' and higher ϵ implies greater generalisation capability of W'. The optimal matrix is then evaluated by extremising $C(\vec{\omega}, \vec{t})$. A more practical version with variable number of computational layers is discussed in the next section.

5.2 Quantum Circuit Implementation of PINN

5.2.1 Construction of W-matrix

The efficient construction of a generalised 2^n dimensional matrix arises from singular value decomposition given by $W_m(\omega_m) = UDV$ where U, V are unitary operators and D is a diagonal matrix defined by $D = diag(d_0, ..., d_{2^n-1})$ with $0 \leq d_i \leq 1 \forall i$. This section attempts to choose how to parameterise the constituent matrices to efficiently construct quantum gates for the general purpose of quantum machine learning signalled by PINN. The individual ϵ_x values are pre-selected based on the problem and hence not varied here. More specifically, we try to design U, V, D with ω_m as the sole parameter. The unitaries U, V are universally constructed from C-NOT gates and 2-dimensional unitary operators(which has an universal construction) and that leaves us with the diagonal matrix. As mentioned in the paper[], X-gate(NOT) and 2^n number of $D_i = diag(1, , .., d_i)$ operators can build up D. Further factorisation of D_i furnishes a $C^{n-1}[N_1(a)]$ gate(controlled $N_1(a)$ gate with $n-1$ control qubits) where $N_1(a)$ is 2 level non unitary gate given by $diag(1, a)$ s.t. $0 \leq a < 1$. The $N_1(a)$ gate is composed of the Projective Measurement operator in the $|0\rangle$ basis $N_1(0) = |0\rangle\langle 0|$ and $U_1(a)$ where;

$$U_1(a) = \begin{bmatrix} a & \sqrt{1-a^2} \\ \sqrt{1-a^2} & -a \end{bmatrix} \tag{22}$$

The matrix is a Rotation operator in 2-D real space. The probability of success of the non-Unitary gate is measured by the creation of a Measurement basis $M_0 = cW, M_1 = \sqrt{I - M_0^+ M_0}$ with final result $|\psi_f\rangle = \frac{W|\psi_0\rangle}{\langle\psi_0|W^+W|\psi_0\rangle}$. A detailed analysis is found in the paper[]. A simple example of FPI formulation is seen if we pre-select $\epsilon = \frac{2\pi}{N}$ and $\omega_{\alpha,\beta} = \alpha\beta$ which leads to the Quantum Fourier Transform as $W_{\alpha,\beta} = e^{\frac{2\pi i \alpha\beta}{N}}$.

5.2.2 Constraint Selection Possibilities

As is shown in the diagram for the general FPI derivation; the starting point marked by a is connected to all the points in the next chosen ϵ separated line. The total time taken to reach the final destination marked by b is reached in finite steps of ϵ. The number of steps is marked by the total set of possibilities allowed in the system. This set of possibilities is marked by various points in space. In our PINN model; the set of possibilities is marked by the independent number of bases in the Hilbert Space. For an abstract model the final point in space-time is the objective of the cost function defined as $C(\vec{\omega}, \vec{t})$. This means that the natural process to learn the entire functional relationship ω_m is carried out in N layers where N is the cardinality of the space. To optimally learn in such a space; we need to select the most relevant features in space-time marked by (ω_m, ϵ_m) by using the cost function.